June [illegible]
US [illegible]

80 microcomputing

the magazine for TRS-80 users

PRINTERS: HOW DO THEY STACK UP?

57 Compared in 80's First Buyers Guide to Printers.

TRS-80* COMPUTING EDITION

©1981 Percom Data Co., Inc. **The Percom Peripheral** 35 cents

Percom's DOUBLER II™ tolerates wide variations in media, drives

GARLAND, TEXAS — May 22, 1981 — Harold Mauch, president of Percom Data Company, announced here today that an improved version of the Company's innovative DOUBLER™ adapter, a double-density plug-in module for TRS-80* Model I computers, is now available.

Reflecting design refinements based on both theoretical analyses and field testing, the DOUBLER II™, so named, permits even greater tolerance in variations among media and drives than the previous design.

Like the original DOUBLER, the DOUBLER II plugs into the drive controller IC socket of a TRS-80 Model I Expansion Interface and permits a user to run either single- or double-density diskettes on a Model I.

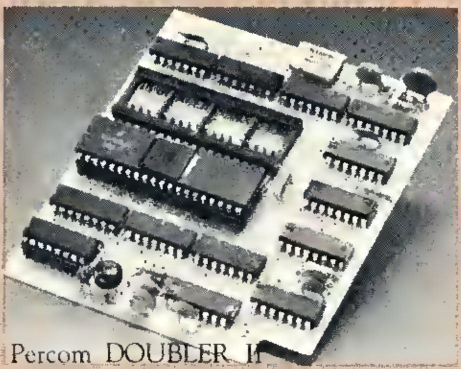
With a DOUBLER II installed, over four times more formatted data — as much as 364 Kbytes — can be stored on one side of a five-inch diskette than can be stored using a standard Tandy Model I drive system.

Moreover, a DOUBLER II equips a Model I with the hardware required to run Model III diskettes.

(*Ed. Note*: See "OS-80™: Bridging the TRS-80* software compatibility gap" elsewhere on this page.)

The critical clock-data separation circuitry of the DOUBLER II is a proprietary design called a ROM-programmed digital phase-lock loop data separator.

According to Mauch, this design is more tolerant of differences from diskette to diskette and drive to drive, and also provides immunity to performance degradation caused by circuit component aging.



Percom DOUBLER II

Mauch said "A DOUBLER II will operate just as reliably two years after it is installed as it will two days after installation."

The digital phase-lock loop also eliminates the need for trimmer adjustments typical of analog phase-lock loop circuits.

"You plug in a Percom DOUBLER II and then forget it," he said.

The DOUBLER II also features a refined Write Precompensation circuit that more effectively minimizes the phenomena of bit- and peak-shifting, a reliability-impairing characteristic of magnetic data recording.

The DOUBLER II, which is fully software compatible with the previous DOUBLER, is supplied with DBLDOS™, a TRSDOS*-compatible disk operating system.

The DOUBLER II sells for $219.95, including the DBLDOS diskette.

Owners of original DOUBLERs may purchase a DOUBLER II upgrade kit, without the disk controller IC, for $30.00. Proof of purchase of an original DOUBLER is required, and each DOUBLER owner may purchase only one DOUBLER II at the $30.00 price.

The Percom DOUBLER II is available from authorized Percom retailers, or may be ordered direct from the factory. The factory toll-free order number is 1-800-527-1592.

Ed. note: Opening the TRS-80 Expansion Interface may void the Tandy limited 90-day warranty.

Circuit misapplication causes diskette read, format problems.
High resolution key to reliable data separation

GARLAND, TEXAS — The Percom SEPARATOR™ does very well for the Radio Shack TRS-80* Model I computer what the Tandy disk controller does poorly at best: *reliably* separates clock and data signals during disk-read operations.

Unreliable data-clock separation causes format verification failures and repeated read retries.

CRC ERROR–TRACK LOCKED OUT

The problem is most severe on high-number (high-density) inner file tracks.

As reported earlier, the clock-data separation problem was traced by Percom to misapplication of the internal separator of the 1771 drive controller IC used in the Model I.

The Percom Separator substitutes a *high-resolution* digital data separator circuit, one which operates at 16 megahertz, for the low-resolution one-megahertz circuit of the Tandy design.

Separator circuits that operate at lower frequencies — for example, two- or four-megahertz — were found by Percom to provide only marginally improved performance over the original Tandy circuit.

The Percom solution is a simple adapter that plugs into the drive controller of the Expansion Interface (EI).

Not a kit — some vendors supply an untested separator kit of resistors, ICs and other paraphernalia that may be installed by modifying the computer — the Percom SEPARATOR is a fully assembled, fully tested plug-in module.

Installation involves merely plugging the SEPARATOR into the Model I EI disk controller chip socket, and plugging the controller chip into a socket on the SEPARATOR.

The SEPARATOR, which sells for only $29.95, may be purchased from authorized Percom retailers or ordered directly from the factory. The factory toll-free order number is 1-800-527-1592.

Ed. note: Opening the TRS-80 Expansion Interface may void the Tandy limited 90-day warranty.

All that glitters is not gold
OS-80™ Bridging the TRS-80* software compatibility gap

Compatibility between TRS-80* Model I diskettes and the new Model III is about as genuine as a gold-plated lead Krugerrand.

True, Model I TRSDOS* diskettes can be *read* on a Model III. But first they must be converted and re-recorded for Model III operation.

And you *cannot write* to a Model I TRSDOS* diskette. Not with a Model III. You cannot add a file. Delete a file. Or in any way modify a Model I TRSDOS diskette with a Model III computer.

Furthermore, your converted TRSDOS diskettes cannot be converted *back* for Model I operation.

TRSDOS is a one-way street. And there's no retreating. A point to consider before switching the company's payroll to your new Model III.

Real software compatibility should allow the *direct, immediate* interchangeability of Model I and Model III diskettes. No read-only limitations, no conversion/re-recording steps and no chance to be left high and dry with Model III diskettes that can't be run on a Model I.

What's the answer? The answer is Percom's OS-80™ family of TRS-80 disk operating systems.

OS-80 programs allow *direct, immediate* interchangeability of Model I and Model III diskettes.

You can run Model I single-density diskettes on a Model III; install Percom's plug-in DOUBLER™ adapter in your Model I, and you can run double-density Model III diskettes on a Model I.

There's no conversion, no re-recording.

Slip an OS-80 diskette out of your Model I and insert it directly in a Model III.

And vice-versa.

Just have the correct OS-80 disk operating system — OS-80, OS-80D or OS-80/III — in each computer.

Moreover, with OS-80 systems, you can add, delete, and update files. You can read *and write* diskettes regardless of the system of origin.

OS-80 is the original Percom TRS-80 DOS for BASIC programmers.

Even OS-80 utilities are written in BASIC.

OS-80 is the Percom system about which a user wrote, in Creative Computing magazine, ". . . the best $30.00 you will ever spend."†

Requiring only seven Kbytes of memory, OS-80 disk operating systems reside completely in RAM. There's no need to dedicate a drive exclusively for a system diskette.

And, unlike TRSDOS, you *can* work at the track sector level, defining and controlling data formats — in BASIC — to create simple or complex data structures that execute more quickly than TRSDOS files.

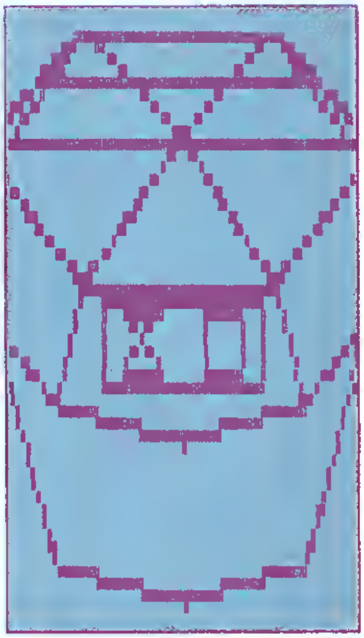
The Percom OS-80 DOS supports single-density operation of the Model I computer — price is $29.95; the OS-80D supports double-density operation of Model I computers equipped with a DOUBLER or DOUBLER II; and, OS-80/III — for the Model III of course — supports both single- and double-density operation. OS-80D and OS-80/III each sell for $49.95.

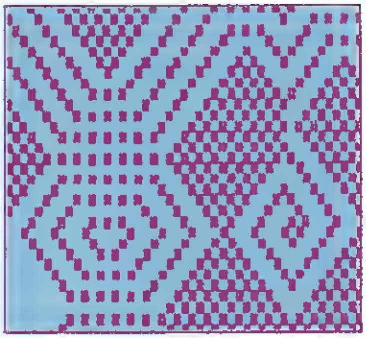
PRICES AND SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE. PRICES DO NOT INCLUDE HANDLING AND SHIPPING.

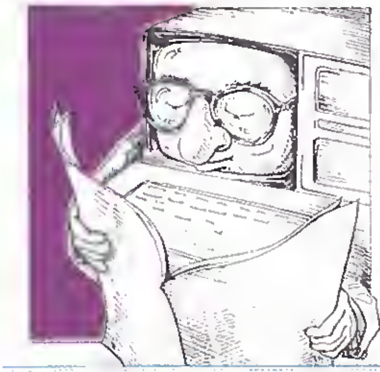
PERCOM DATA COMPANY, INC. 211 N. Kirby Street Garland, Texas 75042 (214) 272-3421 258

™Trademark of Percom Data Company, Inc. *TRS-80 and TRSDOS are trademarks of Tandy Corporation which has no relationship to Percom Data Company. †Creative Computing Magazine, June, 1980, page 26.

80 Contents







80 Microcomputing (ISSN -0199-6789) is published monthly by 1001001 Inc., 80 Pine St., Peterborough NH 03458. Phone: 603-924-3873. Subscription rates in U.S. are $18 for one year and $45 for three years. In Canada, $20—one year only, U.S. funds. Canadian distributor: Micro Distributing, 409 Queen St. West, Toronto, Ontario, Canada M5V 2A5. Foreign subscriptions (surface mail), $28—one year only, U.S. funds. Foreign subscriptions (air mail), $60—one year only, U.S. funds. In Europe contact Monika Nedela, Markstr. 3, D-7778 Markdorf, W. Germany. In South Africa contact *80 Microcomputing*, P.O. Box 782815, Sandton, South Africa 2146. Australian Distributor: Electronic Concepts, Attention: Rudi Hoess, 55 Clarence Street, Sidney 2000, Australia. All U.S. subscription correspondence should be addressed to *80 Microcomputing*, Subscription Department, P.O. Box 981, Farmingdale, NY 11737. Please include your address label with any correspondence. Postmaster: Send form -3579 to *80 Microcomputing*, Subscription Services, P.O. Box 981, Farmingdale, NY 11737. Additional mailing offices at Hollywood, Florida 33020

APPLICATIONS

GAME

GENERAL

GRAPHICS

INTERFACE

PERSONAL

REVIEW

TECHNIQUE

TUTORIAL

UTILITY

DEPARTMENTS

Cover photo by Martin J. Paul

DOSPLUS

FEATURES:

1) Radio Shack compatibility
2) Error free variable length records
3) Full lower case detection and support
4) Repeating keyboard with NO keybounce EVER
5) Shift [0] typewriter keyboard option
6) Execute only protection feature for BASIC programs
7) Automatic track support for 35 through 80 track drives (mixed)
8) Device I/O handling with FORCE command
9) Supports high speed clock modification (up to 4.0mhz)
10) Supports mixed mode (single & double density) automatically
11) Allows disable-enable of break key
12) Allows user to define step rate per drive and re-configure system disk
13) Allows for efficient use of double-headed drives
14) Built in screen printer (shift [CLEAR]) with [BREAK] key abort
15) Multiple command chaining with "DO"
16) Built in memory test with CLEAR command
17) New printer driver which allows complete forms control and paging
18) Automatic serial printer driver with optional auto linefeed
19) Execute any DOS command from BASIC and return to BASIC
20) Free space map of diskette with optional output to printer
21) Copy with variable length files
22) Complete RS232 control from keyboard with status check
23) Create and pre-allocate files from DOS
24) Display current date and time from DOS
25) More information from Directory with optional printer output
26) Enter DEBUG with shift [BREAK] to allow use of [BREAK] from BASIC
27) New DISKDUMP/CMD sector display/modify program (works with filespecs)
28) New DISKZAP/CMD single/double density disk editor
29) New BACKUP (more reliable, no more pack ID check)
30) New FORMAT (more reliable, no need to bulk erase disk first)
31) New MAP utility (maps out disk, showing where files are located)

New DOSPLUS Z80 Extended Disk BASIC

1) Faster loads and saves
2) BASIC Reference utility (lines, variables, keywords, printer option)
3) BASIC Renumber utility (renumber section of text, block text move)
4) Shorthand features for almost ANY direct command (LOAD, SAVE, etc.)
5) Shorthand-features for editing (listing and editing with single key)
6) CMD"M" instantly displays currently set variables
7) Global search and replace in BASIC text
8) Line printer TAB to 255
9) OPEN"E" to end of sequential file (for output)
10) DI (delete and insert text line)
11) DU (duplicate text line)
12) ",R" & ",V" options after LOAD and RUN (files open & save variables)
13) OPEN"D" allowed (Model II compatible) equal to OPEN"R"
14) DOS commands from BASIC
15) Automatic, error-free variable length records
16) Single step execution with TRON (fabulous for debugging)
17) CRUNCH (BASIC program compressor)
18) New TBASIC (tiny BASIC) offers full BASIC commands
19) TBASIC and DOSPLUS together only use 8K of RAM (40K left in 48K TRS-80)

***** 7 MORE UTILITIES *****

1) Single drive copy
2) Restore (dead files)
3) Purge (unwanted files)
4) Clearfile (destroys data by writing zeros to file)
5) Transfer (moves all user files from one disk to another)
6) Spooler (allows printing of text while freeing up the CPU)
7) Crunch (Basic program compressor)

***** ALSO *****

- New I/O package 30% faster
- No BREAK key death from DOS
- No closing killed files and ruining diskettes

DOSPLUS gives you more of what you buy an operating system for. Speed and reliability without sacrificing simplicity and power. If you need extra power without extra wait, then you need DOSPLUS!

Single or double density systems available for Model I. Model III DOSPLUS ready for immediate delivery.

Perhaps the best investment you can make for your TRS-80! Listen to what others have had to say about DOSPLUS.

"Overall, DOSPLUS is the fastest operating system I have seen..."
Pete Carr in 80-US Journal.

"DOSPLUS...the better mousetrap."
Stewart Fason in 80-Microcomputing

"On a scale of 1 to 10, I give DOSPLUS a solid 9."
Reese Fowler in 80-Microcomputing
(Model III DOSPLUS review)

For the BASIC programmer, our features are unmatched. For the average businessman, our speed and simplicity cannot be beat.

So, join the satisfied users who have joined DOSPLUS. Experience excellence! Experience DOSPLUS!

DOSPLUS comes complete with full utilities, PLUS a FREE patch to enable Model I Scripsit/Super Script to run on Model III, UNLIMITED Backups!

Model I DOSPLUS — $99.95
Model III DOSPLUS — $99.95
Model I double density upgrade — $175.00
Master Directory 1.2 (double density) — $29.95

STEP ON THOSE DOS BUGS!! ORDER TODAY!!

VISA

master charge

Micro Systems Software inc ← 384
Specializing in the Tandy Line

MS inc.

(305) 983-3390
5846 Funston Street
Hollywood, FL 33023

CALL TOLL FREE FOR FAST SERVICE
(800) 824-7888, OPERATOR 193
FOR VISA/MASTERCHARGE/C.O.D. ORDERS
California dial (800) 852-7777, Operator 193
Alaska and Hawaii dial (800) 824-7919, Operator 193
TOLL FREE LINES WILL ACCEPT ORDERS ONLY!
For Applications and Technical information, call (305) 983-3390 or drop us a card.

Dealers inquiries invited

80 REMARKS

by Wayne Green

"The new magazine (Desktop Computing) will have articles on the successful business uses of small computers."

The Pocket Computer

Despite the lack of consumer enthusiasm for the Pocket Computer, Radio Shack has not lost faith. We are promised memory expansion, a printer and other goodies as support.

The noticeable absence of user interest in the PC has kept us from doing much for the system in this magazine. Perhaps many PC owners reacted as did I—becoming a bit disillusioned at the poverty of even remotely useful programs in the thick book that accompanies the computer. I was further put off by the miniscule internal memory and the difficulty involved in expanding that memory. I hoped that some of our readers would grab the PC and quickly send us some articles on memory addition or interfacing to the Sharp Memowriter. Nothing much has happened. . . so, folks, get on the stick!

"Since the main benefit of the PC is its very small size, ways of adding memory without significantly increasing the size will be welcome."

With some memory expansion, a small word processing program and a miniature TV you could have a traveling typewriter which would dump the material onto a cassette, much like the Sony Typecorder, but with more flexibility and lower cost.

Since the main benefit of the PC is its very small size, ways of adding memory without significantly increasing the size will be welcome. Even a system which will fit in a thin attache case would be desirable.

The Color Computer

Another Radio Shack computer which has gotten little play so far is the color system. With the total lack of available software, there is no reason for any businessman to even take a look at it. But since it seems likely that this system is going to be around for a while, I'd like to see some articles on it. We want to know what changes have to be made in BASIC programs to get them to run on the color system. We want to know more about color graphics and how to get into the machine to make these graphics more useful and flexible. We want to see programs for it, as well as successful conversions.

Then there is the Model III, which is fairly compatible with Model I programs. We all want to know more about its differences and what to do about them. If you've had to change any of the published programs in *80* to get them to run on the Model III, let us know about it.

The New Magazine

To provide businessmen with a magazine which will tell them what computers are doing now and will do in the future, we're planning a magazine for fall debut which will be written in plain English.

The new magazine (*Desktop Computing*) will have articles on the successful business uses of small computers. Since you are in touch with many of the businesses where desktop computers are being used, this is another publishing opportunity for you. Readers of the new magazine will want to know what system a business chose, some of the reasons why, what hardware, software and accessories were used, what problems were encountered and how they were solved, and, most importantly, the benefits of the new system. Did the installation save money, eliminate a job?

The tricky part of writing for *Desktop Computing* will be avoiding terms with which businessmen are not familiar. You may know just what is meant by a global search and replace, but the average businessman will be bewildered.

Royal Shafting?

Some wag apparently sent my name to Royal as a prospect to buy a word processing system from them. The literature package arrived, and I do want to thank whoever did me that good turn. Not that I became interested in the Royal system, but rather that I was surprised at how little computer you get for $11,500.

Royal no doubt has a very nice word processor. But it is dedicated and, from the literature, can be used for nothing else. Compare that to the TRS-80 word processor, where, with the flick of a disk, you can use the same system for bookkeeping, inventory, invoicing and making sales charts. And it costs about one third the price of the Royal. How can they stay in business?

"We want to know more about color graphics and how to get into the machine to make these graphics more useful and flexible."

Do you realize how much computing equipment you can buy for $11,500? I think I could outfit a small office for the same price as one dedicated Royal word processor. Buyers beware! ■

***Notice:** Because of a late mailing William Barden's column, The Assembly Line, will not be appearing this month. Our apologies. He'll be back next month.*

INSIDE 80

by Ed Juge, director of computer merchandising, Tandy Radio Shack

"If you're really knowledgeable. . .in a viable field, and you believe you could define a really outstanding use for the TRS-80 in that field, write to us."

Last week, Bill Walters (our Consumer Information Manager) and I visited *80 Microcomputing*'s nice folks in beautiful Peterborough, N.H. Know what? Those employment ads Wayne runs every so often don't lie—it's a lovely place! It's also true that it's a non-smoker's place to work. Since Bill and I are both successfully reformed smokers, we were allowed in for a look around and a very enjoyable day. We found the staff hardworking, enthusiastic, bright and very friendly. If you don't smoke and think you're either young enough or crazy enough for this sort of work, I'm sure they'd like to hear from you.

If you do smoke (or even if you don't. . . we're open-minded) keep Radio Shack in mind too! Our R&D group keeps an eye open for talented engineers, systems and applications software analysts and programmers. Our documentation group is always on the lookout for people to develop manuals for our software and hardware products. There are also two software testing groups, one in the software area, and one in our Merchandising Department.

There are long hours, great people to work with, and some outstanding fringes. If you're interested, drop a letter and resume to Mr. George Berger, our Personnel Director at 500 One Tandy Center, Ft. Worth, TX 76102.

Model II Scripsit Owners

While I'm enjoying Scripsit on my Model II right now, I'll pass on some information. We have received comments from a number of people about funny things happening during printout: pages printed with only one or two lines on them, or even completely blank pages printed between correctly formatted ones. As far as I know, there is one thing you can do which will cure all these problems—before you print a document, issue a repaginate command. This is especially important if you've added text after a previous repagination.

Repagination adjusts your pages to the maximum allowed number of lines. If you then add a couple of lines, that page is too long. The extra lines are stored in the proper place, but can only print as an additional (and usually unwanted) page. To make room for the addition, all following pages must be readjusted. We left that for you to do manually, saving time when you're editing a multi-page document.

Wanted. . . A Variety of Experts

There are lots of vertical markets (uses which are specific to a particular kind of business) for computers. Unfortunately, we haven't found a way to become experts in all business and professional fields. . . yet there is a great demand for software for specific vertical markets. We'd like to provide for as many as we can. Maybe there are a lot of you who could help us and yourselves.

If you're really knowledgeable, or a recognized expert (whatever that means) in a viable field, and you believe you could define a really outstanding use for a TRS-80 in that field, write to us.

We work with outside people on about 60 percent of our software, so it's not a new idea, just not too well known. Now that our basic accounting and word processing packages exist, the vertical markets are most interesting to us.

Complaints, Again

Complaints and misunderstandings (in plain language, our hate mail) have reduced significantly in recent months. I still would like to touch on some complaints or misconceptions from time to time, just for those who might not have heard the answer to their specific questions.

Our computers are designed for the needs of the target market: cost effectiveness, ease of repair, etc. We don't trade off these items just so they'll easily interface with existing peripherals. Neither do we make an effort to make interfacing difficult. Come on, guys, we have to interface them to the outside world too, when we produce our own peripherals.

Want information? Phone calls won't often get the technical information you want. However, I am assured that our engineering group (which includes systems software) will answer almost any technical question (we won't do custom engineering or software for you) which is submitted by mail. They will answer specific questions about, but will not furnish source code listings to, our systems software. You should be able to get any answer for which you know how to ask the question. If you find my information to be untrue, I'd like to hear about it, and who you corresponded with.

Warranty, Warranty. . . Who's got the Warranty?

We've received many questions recently about who to buy your TRS-80 from, and whether or not you get a warranty. We hear that a few of our salespeople may be leaving the wrong impression with customers as to company policies, warranties and dealers, so I'd like to clear some of the fog.

Radio Shack has company owned stores, as well as authorized dealers and franchisees. There is absolutely no difference in the Radio Shack merchandise you get from these outlets, or the warranty you're entitled to on it. We control our stores, while authorized dealers are independent businessmen who buy our products and resell them to you. Their operating policies and how they run their business is up to them. If you buy any Radio Shack product from any of our authorized outlets, you can return it to any Radio Shack store anywhere, for warranty service. You *will* need your original sales ticket to secure in-warranty service.

Although I'm often asked, I'm not going to tell you where to buy, because it shouldn't matter. I will make one common sense suggestion. Anything electronic can go wrong; when you need help or advice, you're almost always going to fare better with the guy who made the profit on the sale.

Radio Shack requires its company owned store personnel to handle any customer's problem in a timely fashion. Human nature simply says that a company store employee or dealer is much less likely to give up his lunch hour to help

Continue to page 39

80 INPUT

". . . accusing fingers have been pointed toward the poor TRS-80 and its apparent inability to handle tabs in excess of 64."

HOWZAT?

While I appreciate your publishing my article ("WHERZIT") in the April issue, your choice of a title is a cruel joke, given the article's premise. Please assure your readers that I had no part in its selection.

There is a typo on page 254 of the article that we all missed. CLEAR 44000 is correct for a *48K* machine, not a *64K* machine.

James H. Fox
Afton, MN 55001

You're no fun!—Eds.

DOS vs DOS

Mr. Fason's review of Percom's doubler was very interesting to me, and I certainly agree that the product is a boon to disk storage space. The DBLDOS sale literature and the review, however, are very misleading; it does support TRSDOS commands, but only those of version 2.1. If one is fortunate enough to have version 2.3, you learn that you not only lose DEVICE, which is not loss in most cases, but you also lose BASICR and COMMAND "I". Now, that is very disappointing, especially when one has hopes of doing some long postponed MERGing.

I would appreciate any assistance you can give in the way of an article or source of a technique to get TRSDOS version 2.3 into DBLDOS.

Joe Restle
Langhorne, PA 19047

Double Density DOSPLUS from Micro Systems Software is a better alternative to DBLDOS and is a more powerful DOS than even TRSDOS 2.3.—Eds.

80 Tabs

For many months now, accusing fingers have been pointed toward the poor TRS-80 and its apparent inability to handle tabs in excess of 64. I suggest that the accusers look to the real culprit, their printers.

I use a NEC Spinwriter 5330 which accepts tabs from 0 to 132 without a murmur; tabs in excess of 132 executes a CR/LF. I do not know what the upper limit is, but would hazard a guess at 65535.

Big tabs work equally well with the new two-chip ROM and the older three-chip ROM when used with an intelligent printer. So stop blaming the 80, there are very few shortcomings with it, and most of these can be attributed to software and peripherals.

C. J. Casselden
Sutton, Surrey, England
SM2 5DL

Easy Machine Language

My eternal gratitude for Lt. John Harrell's article (January 1981) on the Super Bug Monitor. As the proud new owner of a TRS-80 Model III, I was eager to begin writing machine language. To my chagrin, Radio Shack had no EDTASM for the Model III. I had written a BASIC program to permit me to input hex into RAM, and was blindly POKEing in machine code.

Upon reading Lt. Harrell's article and perusing the listing for his program, I figured the monitor would work in the Model III as long as the ROM calls would work. The ROM routines listed in the Model III Manual were the same as Lt. Harrell's, so I hoped the others would be, also.

Lo and behold! All ROM calls worked. The program works *exactly* as described.

After loading and testing the program in high RAM (6330H), I relocated it to where Lt. Harrell had located it (4330H). However, some of the code kept getting overwritten; apparently the Model III uses 4420H-4430H as some kind of stack.

I now eagerly await an assembler-disassembler for the Model III. Lt. Harrell's monitor has eased the pains of machine language programs, but I am still manually assembling them. Perhaps a vendor has an assembler for the III? (If so, none indicate such in any ads in your great mag.)

Bruce C. Hampton
320A N. Kenwood
Glendale, CA 91206

Thanks Readers

Thank you very much for sending our Folsom Prison Computer Group the complimentary subscription to *80 Microcomputing*, and for publishing my letter in your February issue. As a result of the letter, many of your readers sent our group a fine selection of software tapes, textbooks, and various computer related materials.

In January, our Education Department initiated a course in Introduction to Microcomputers, which will be followed by a course in BASIC language for micros. So, our program is starting to move at a faster pace and we will be able to make good use of all the materials sent to us by readers of *80 Microcomputing*.

On behalf of our computer group, I would like to take this opportunity to thank all of the many readers who sent materials to us, and to again thank you and your staff for your interest and assistance. All of us who will be making use of the materials sincerely appreciate the efforts extended on our behalf.

Gottfried R. von Kronenberger
P.O. Box B-49542
c/o Mr. R. E. Miller,
Supervisor of Education
Folsom State Prison
Represa, CA 95671

Pennington Pal

The following letter was addressed to H. C. Pennington.

Congratulations on a very fine article in the March issue of *80 Microcomputing*. I wish more people realized what we have gone through with the TRS-80. I read John Grass's letter and it became obvious to me that he was naive as to what makes a disk operating system work. I'm glad you wrote the reply.

I am very skeptical of anything that Radio Shack sells. They just don't quality control their products very well before they are released to the public.

As a design engineer, I don't think much of their hardware design. The tape interface circuitry in particular amazes me that it works as well as it does.

I have your book and am very, very pleased with it. That kind of information is not commonly known. I, like John Grass, am also looking for a good book disassembling Level 2 and Disk BASIC. If you have a specific recommendation, I'd appreciate hearing about it.

John Zdenek
Riverside, IL 60546

Letter vs Book

If I were Pennington and English were my fourth language, I would still be embarrassed by the semiliterate style, or lack thereof, of *TRS-80 Disk and Other Mysteries*. Had he written his book as well as the rebuttal of Mr. Grass' letter, it would have been a masterpiece.

Louis Zeppa
Sacramento, CA 95819

European Orders

I have unfortunately had a bad experience with some American firms who regularly advertise in your magazine, but who pursue a most dubious business policy vis-a-vis European customers.

For example: In October 1979 I ordered a one year subscription from H & E Computronics. As demanded, I paid in advance by bank transfer. In spite of several letters and reminders, this firm seems not to be willing to carry out my order. Meanwhile I have asked the German Consulate in New York for help. But, despite telephone calls and reminders, even the Consulate had no success. In their letter of November 12, 1980 they wrote: "Leider hinterließ die Firma auch beim Generalkonsulat keinen guten Eindruck," which means, "This firm did not leave a good impression vis-a-vis the General Consulate."

In order to avoid such losses to other readers of your magazine, I would be glad if you published this letter.

I had the same experience with Cost Effective Computer Service, who received my money order of September 19, 1980 for a TRS-80 program, which they have not delivered.

Claus Behnke
Dasnoeckel 59
D-5600 Wuppertal 11
W. Germany

Unfortunately, Mr. Behnke does what many other foreign buyers do. Mr. Behnke had his bank (or post office) forward a check to a United States company. When these checks arrive, they come without any explanation and usually contain an incomplete address. In the case of Mr. Behnke, we received a check without any explanation. The address that appeared on the check was: Desnoeckel 59, 5600 Wuppertal 11.

We didn't even know what country the check came from and had no way to contact the customer.

Our policy is to deposit these checks after making a photocopy of the check. We always have a file of about 20 outstanding checks of this kind. We have to wait for the customer to contact us (usually stating that they had sent us a check and didn't receive their order). At that time, we check the customers complaint against our outstanding check file and try to find a check that matches the customer's order.

We suggest that you advise all 80 Microcomputing *readers that they should always send their orders with their check.*

Maralin, Customer Service
H & E Computronics

Ed's Note: A copy of Mr. Behnke's complaint was also sent to Cost Effective Computer Service with an offer to print any reply they might care to make. As of this date, no reply has been received.

Character Generator Works

I would like to clarify and update a few points brought up in Eric Keener's letter on page 16 of the March issue.

The character generator IC, part number AXX-3027, whose catalog number is 26-1104, cannot be ordered from a Radio Shack store. It can only be obtained through the Service Department. The address has been changed to: Radio Shack Customer Service #0048, 900 E. Northside Dr., Fort Worth, TX 76102. It can also be ordered by calling 817-870-5662. I used the latter method and received the part in seven days.

I installed the lowercase modification that was published on page 72 of the March 1980 issue. I was hesitant about ordering the IC because a friend of mine had the Radio Shack version installed, which was wired slightly differently than the one I had installed. I decided to go ahead and buy it even though I could not be certain it would work. It arrived after seven days.

I opened the case of my computer and examined both chips carefully to see if there was much difference in the part number. The one in my computer had the number 8046670 and the new chip had the number 8046673. Figuring that there wasn't much difference, I replaced the old with the new and put my computer back together. I reconnected everything and powered it up. A-OK, DOS booted up and everything seemed normal. I turned on the lowercase switch and ran SCRIPSIT/LC. Voila, I had normal uppercase and lowercase with descenders. It works fine for me.

Jeffery A. LeBlanc
548 Marion Ave.
Lima, OH 45801

Airborne Computer Programs

I am producing a weekly communications magazine program in English for the Dutch External Service, which is broadcast on short-wave to a world-wide audience each Thursday. At the suggestion of some of our listeners in the USA, Europe and the Pacific, we intend to try an experiment fairly shortly, which might be of interest to your readers.

On Thursday September 10th, 1981, we will be devoting our Media Network program to the subject of home computers and how they can be of use to the active short-wave listener. As well as an introduction to microcomputing, we will also be including a short computer program in three different formats, broadcast in machine readable form over the air. Providing the signal strength is sufficient in the listener's area, we hope it will be possible to record the computer program (off the air) onto cassette tape and play it back into a home computer. Preliminary experiments indicate that the system should work, but the purpose of the experiment on September 10 is to gauge whether atmospheric noise is low enough in most of our target areas to enable the scheme to work. If successful, the idea might be repeated on a more regular basis.

Three computer programs will be transmitted, of use to the short-wave listener, to be compatible with Tandy Radio Shack, Apple, and Commodore PET microcomputers.

The time chosen is the beginning of the program Media Network, which runs for 30 minutes. All times are quoted in Greenwich Mean Time, which is five hours *ahead* of Eastern Standard Time.

Listeners who hear the broadcast, and try out the computer program, are encouraged to write in and report their results to the following address:
Computer Experiment
Media Network
Radio Netherlands

Continued to p.14

P.O. Box 222
1200 JG Hilversum, Holland

J. Marks, Producer
Media Network
Radio Netherlands
P.O. Box 222
1200 JG Hilversum, Holland
Tel: 16151
Tlx: 43336

Eds. Note: See box below for broadcast times and frequencies.

Mod III List

I am attempting to compile a list of TRS-80 Model I programs that will (or will not) run on the Model III. I would appreciate any input from your readers.

To those who kindly respond, please mention if the program was on disk or tape—BASIC, System or disk CMD file—whether run under Level II or DOS, and, if DOS, which one. If any changes needed to be made to the program, what were they?

All those who send me information will be sent the compiled list (after a reasonable length of time to get all input) *if* a SASE is included with your information.

Ken Knecht
1340 W. 3rd St. #130
Yuma, AZ 85364

Business and Model I

I quite agree with Dr. Goldstone in his letter "Radio Shack and Model II" in the April issue, p. 21.

The very reason I chose and purchased the TRS-80 Model I was due to its modular design. I certainly didn't purchase it for its good looks. Had I been interested in a good-looking computer I'd have gone elsewhere. Instead, I wanted one that was both functional and convenient. It will not be at all convenient for Model II and Model III owners to interrupt their computing because they have to haul the whole thing in for repairs. I'm not saying that those Models don't have a market (obviously they do). The same ought to be true for Model I. I compare it to the purchase of fine stereo equipment—modular systems are much more preferable to the music enthusiast.

Perhaps Radio Shack is overlooking the continued marketplace for the "ugly duckling" Model I.

Question: Why has the Model I been withdrawn from production? Let's hear Tandy's side of the picture.

Patrick T. McArron, President
Advance Weekends, Inc.
Santee, CA 92071

Rummy Master Notes

Here are some comments on a game I purchased. The name of the program is Rummy Master, by Dave Gubser, and is published by Quality Software. The program is supposed to accept light pen input. However, if a light pen is used, an OM error occurs during most games.

There are also two occasions when an SN error will be generated: in line 254, when an ace is melded to an existing 2-3-4 meld, and when a flush meld is attempted with card suits that don't match.

The first bug can be fixed by changing MF(F9) = INT – 130 to MF(F9) = CA – 130. The second bug can be fixed by changing :3: to :GOSUB 3: and retyping the rest of the line using the hack and insert edit command.

One other problem is that the computer will not allow you to call Rummy. This problem I have not been able to fix. . . yet. If you call Rummy while using the light pen option, the program crashes back to the beginning of the game.

If Rummy is called while using the keyboard, the computer just ignores you. If anyone has a debug for this problem I would like to hear from you.

There are two cures for the OM error problem: Add more memory to your computer, or make two programs out of the one, separating the light pen and the keyboard options.

I hope this helps others who have this program.

John F. Costello
Philadelphia, PA 19127

Likes Hayden

A public thank you and "attaway" to the Hayden Book Company.

Several months ago I purchased Sargon II and enjoyed many hours with this fine chess program. My only complaint was that the volume setting of my cassette player was very critical.

After a period of disuse, the tape would not load at any volume level. With fingers crossed, I sent the tape to Hayden. I promptly received a letter of apology from Stephen Radosh, Games and Entertainment Software Editor, along with a new tape that loads correctly.

At $30, Sargon II is a bargain, and Hayden is to be commended for standing behind their product.

James L. Price
Modesto, CA

Patches from Holland

After reading the article "#26-2202 Review" by G. F. Stevens in the February 1981 issue of *80 Microcomputing*, I decided to purchase the package. Much to my surprise, the complete package cost me less than the cassette EDTASM that I purchased two years ago.

The first thing I had to do was fix the crash that occurred when exiting EDIT and M80. I would like to share my fix with your readers.

TRSDOS 2.1/2.2/2.3 does not initialize the Debug jump vector at 4315 (hex) to the correct value at Boot. Both EDIT and M80 prevent a jump to Debug by filling location 4315 (hex) with 00 upon entry. But both programs store the value C3 (hex) at this location upon exit to DOS. Since the jump vector has not been initialized to the correct value, and since the user is probably

Continued to p.16

GREENWICH MEAN TIME		SHORT-WAVE FREQUENCY (kHz)	TARGET AREA
07.47		9770, 9715	Australasia
08.47		9715	Australasia
08.47		15560, 11930, 9895, 6045, 5955	Europe
13.50		17605, 11930, 9895, 6045, 5955	Europe
14.47		11735, 15560, 21480	South-East Asia
18.47		15220, 6020	East Africa
20.47		21685, 17695, 17605, 15220, 9715	West and Central Africa (frequencies also audible in Europe)
02.47	(Note, this and the next trans-	9590, 6165	Eastern North America
05.47	mission are shown as early Friday morning GMT, but it is still Thursday evening in the target area).	9715, 6165	Western North America

Times and frequencies for shortwave programs to be broadcast from Holland, Thursday, September 10, 1981.

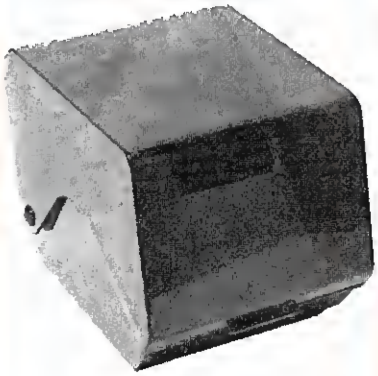
META TECHNOLOGIES

MTC

25111 Brush Avenue, Euclid Ohio 44132
CALL TOLL FREE 1-800-321-3552 TO ORDER
IN OHIO, call (216) 289-7500 (COLLECT)

FILE BOX

DISKETTE STORAGE SYSTEM



$24.95 for 5¼" disks

$29.95 . . . for 8" disks

MTC brings you the ULTIMATE diskette storage system, at an affordable price. Storing 50 to 60 diskettes, this durable, smoke-colored acrylic unit provides easy access through the use of index dividers and adjustable tabs. Unique lid design provides dust-free protection and doubles as a carrying handle.

'RINGS' & THINGS

Help prevent data loss and media damage due to improper diskette centering and rotation with the FLOPPY SAVER™ reinforcing hub ring kit. 7-mil mylar rings install in seconds. Kit is complete with centering tool, pressure ring, 25 adhesive backed hub rings and instructions.

HUB RING KIT for 5¼" disks. $10.95
HUB RING KIT for 8" disks. $12.95
REFILLS (50 Hub Rings) . . . . $ 5.95

Protect your expensive disk drives and your valuable diskettes with our diskette drive head cleaning kit. The kit, consisting of a pair of special "diskettes", cleaning solution and instructions, can be used for 52 cleanings. Removes contamination from recording surfaces in seconds without harming drives.

CLEANING KIT for 5¼" drives . . . $24.95

PLASTIC LIBRARY CASES

(not shown)

An economical form of storage for 10 to 15 diskettes, and is suitable for your bookshelf! Case opens into a vertical holder for easy access.

5¼-inch diskette case $3.50
8-inch diskette case $3.95

Single Sided, Soft-Sectored 5¼-inch, (for TRS-80™) Mini-floppy

DISKETTES

$21.95 box of 10

These are factory fresh, absolutely first quality (no seconds!) mini-floppies. They are complete with envelopes, labels and write-protect tabs in a shrink-wrapped box.

PLAIN JANE™

DISKETTES

The Beautiful Floppy with the Magnetic Personality™

Thousands of people have switched to this low-cost alternative. These quality diskettes are packaged in a plain white box . . . no fancy printing, fancy names or fancy labels, not even our own (labels cost money). Trust us.

PLAIN JANE™ Diskettes $21.95
10 boxes of 10 . . . . (each box)$21.50

PLAIN JANE™ *Gold*

Introducing MTC's premium generic diskette. Single-Sided, Soft-Sectored, DOUBLE-DENSITY, 5¼-inch diskettes with reinforcing HUB-RINGS. Individually 100% ERROR-FREE certified. Invest in GOLD!

PLAIN JANE™ *Gold* $25.95

VERBATIM'S PREMIUM DISKETTES AT AFFORDABLE PRICES

DATALIFE™

Seven data-shielding improvements mean greater durability and longer data life. These individually, 100% error-free certified diskettes feature thicker oxide coating, longer-lasting lubricant, improved liner, superior polishing and more! Meets or exceeds IBM, Shugart, ANSI, ECMA and ISO standards. Reinforcing HUB RINGS help prevent data loss and media damage, reducing errors.

Buy the best . . . buy DATALIFE™

VERBATIM DATALIFE™ DISKETTES
5¼-inch (box of 10)
MD525-01 . . . . $26.95
10 boxes of 10 . . . . (each box)$25.95

8-inch FLOPPIES
Double-Density, FD34-8000 . $43.95

MICROPARAPHERNALIA

DISKETTES (box of ten)

5¼" PLAIN JANE™ $21.95
5¼" PLAIN JANE™ *Gold* $25.95
5¼" DATALIFE™ MD 525-01 . . $26.95
8" DATALIFE™ FD34-8000 . . . . $43.95

NEWDOS by APPARAT

NEWDOS/80 by Apparat $149.95
NEWDOS + to
NEWDOS/80 UPGRADE . . . . . CALL
NEWDOS + with ALL UTILITIES
35-track $69.95
40-track $79.95

BOOKS

TRS-80™ DISK
AND OTHER MYSTERIES . . $19.95
MICROSOFT™ BASIC DECODED $29.95
1001 THINGS TO DO WITH YOUR
PERSONAL COMPUTER . . . . $ 7.95

Let Your TRS-80™ Test Itself With

THE FLOPPY DOCTOR & MEMORY DIAGNOSTIC

by THE MICRO CLINIC

A complete checkup for your Model I. THE FLOPPY DOCTOR completely checks every sector of 35- or 40-track disk drives. Tests motor speed, head positioning, controller functions, status bits and provides complete error logging. THE MEMORY DIAGNOSTIC checks for proper write/read, refresh, executability and exclusivity of all address locations. Includes both diagnostics and complete instruction manual.

SYSTEM DIAGNOSTICS $19.95

An improved version of the SYSTEM DIAGNOSTICS above. Designed for single or double density, 35-, 40-, 77-, or 80-track disk drives. Includes new and modified tests. Features THE FLOPPY DOCTOR, Version 3.0.

SYSTEM DIAGNOSTICS-V3 . . $24.95

MORE PRODUCTS

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation. DATALIFE is a trademark of VERBATIM. PLAIN JANE, AIDS-I, AIDS-III, CALCS-III, CALCS-IV, MERGE-III are trademarks of MTC.

✓20

MOST ORDERS SHIPPED WITHIN ONE BUSINESS DAY

Products damaged in transit will be exchanged.

PRICES IN EFFECT June 1, 1981 THRU June 30, 1981.

Prices, Specifications, and Offerings subject to change without notice.

8106

WE ACCEPT
- VISA
- MASTER CHARGE
- CHECKS
- MONEY ORDERS
- C.O.D.

- Add $3.00 for shipping & handling
- $3.00 EXTRA for C.O.D.
- Ohio residents add 6½% sales tax.

80 DEBUg

OPINION = OOPS!

The following Debug is submitted concerning the article in the March 1981 issue titled "OPINION = PEEK (MAIL)" (page 248):

I fully expect that in 1981 we will see a full network type data base management system released 'for' (not 'by') the Shack to revolutionize information management.

My apologies to any Shack Dealers who might have been bugged by inquiries about the data base manager. And a special apology to Micro Data Base Systems, Inc., of Lafayette, Indiana, which did release this remarkable piece of programming.

Jim Glosser
1425 Eden Rd.
York, PA 17402

Watt Bug?

Re: "Watt's It All About?" on page 219 of the March 1981, *80 Microcomputing*: I found a bug in the program that will result in an incorrect readout at times. I've debugged the program again and offer the following changes that I believe will correct the problem. Also, I added two lines that make the program easier to use. Make the following changes or additions:

Change the following lines:

```
110 and 640—INPUT "For Months June Thru Sep,
  Type 1, Else Type 2"; M1
270  IF M1 = 2 Then 285
443  If R$ = "Yes" or RS = "Y" Go To 93
530  Rem N = Cost Per Each KWH For Next 900
     KWH—Jun Thru Sep
```

Add the following lines:

```
675  INPUT "Another Run Yes or No";R$
680  CLS: If R$ = "Yes or R$ = "Y" Then 600 Else 444
```

Donald W. Hubert
613 Hartless Court
Hampton, VA 23669

KBEEPFIX FIX

I have found the following error in my article, "KBEEPFIX Revisted," in the March 1981 issue. In the third column on page 271, the two lines of machine and assembly language code should be:

Location	Hex Code	Instruction
7F8A	3E 89	LD A,89H
7F8C	32 49 40	LD (4049H),A

Derrell R. Whitehead
11 Petterson Road
Bedford, MA 01730

Lost POKE

Re: "Where Have all the GOTO's Gone?," March, 1981: One command is missing in the printed version of my article.

Under "APPENDING," page 237, top left, following POKE 16548,233, add this command: POKE 16549,66.

Hubert C. Borrmann
2840 S. Circle Dr. #209
Colorado Springs, CO 80906

still pressing the Break key upon entering DOS, a crash may occur.

TRSDOS users should always type DEBUG (OFF) and DIR after a Boot to initialize 4315 (hex) to the correct value. But M80 and EDIT may also be patched to prevent a crash. This is done by filling 4315 (hex) with 00 instead of C3 (hex) upon exit.

Also included is a patch to have EDIT echo a graphic character for the Break key, instead of a dollar sign.

The relative sector of the disk file may be used when patching the programs with Superzap 2.0.

Tom de Man
P.O. Box 169
2250 AD Voorschoten
Holland

PROGRAM NAME	LOCATION IN RAM	OLD VALUE	NEW VALUE	REL. SECT.	BYTE IN SECT.
M80	914EH	C3H	00H	64D	4EH
EDIT	6C65H	C3H	00H	26D	01H
EDIT	6620H	24H	8AH	20D	74H
EDIT	85E1H	24H	8AH	51D	B1H
EDIT	8781H	24H	8AH	53D	59H

Patches from Holland.

BASIC Business

How many times have you heard people say that BASIC is too simple and too slow a language to use for business purposes? I think the problem is that too many have not had the chance to really learn BASIC and the methods that can be used to speed things up in order to use it effectively. The following simple program demonstrates that BASIC is not so bad. The starter program for assembly language seems to be a program to white out the screen since BASIC takes so long. Try this one for speed: It's slower than machine language but . . .

```
10 CLEAR 1000
20 DEFSTR X
30 X=STRING$(255,191)
40 X1=STRING$(4,191)
50 CLS:PRINTX;X;X;X;X1;
60 FOR N=1 TO 2000
70 NEXT N
```

Don't forget the commas.

I am enclosing a sample program showing how we have put this concept to work in doing our own logo.

Peter G. Dunn, President
Sturdivant and Dunn, Inc.
Conway, NH

```
10   CLEAR 3000:DEFSTRX:GOSUB500:/COPYRIGHT
     1980 BY PETER G. DUNN
20   FOR N=1 TO 2000:NEXTN:END
500  XC=CHR$(191):XD=STRING$(56,191):XE=
     STRING$(8,32)
510  XA=XD+XE+STRING$(12,191)+" STURDIVANT
     "+XC+" AND "+XC
520  XA=XA+" DUNN, "+XC+" INC. "+STRING$
     (11,191)+XE+XD
530  XB=XE+STRING$(2,191)+" SMALL "+XC+"
     BUSINESS "+XC
540  XB=XB+" SOFTWARE ::+XC+" FOR "+XC+"
     THE "+XC+" EIGHTY's "
550  XB=XB+STRING$(2,191)+XE+XD
560  X="THIS IS THE BOOKKEEPER":XF="JUST A
     MINUTE":CLS
570  PRINT@324,XA:PRINT@508,XB:PRINT@725,X:
     PRINT@857,XF:RETURN
```

META TECHNOLOGIES

26111 Brush Avenue, Euclid Ohio 44132
CALL TOLL FREE 1-800-321-3552 TO ORDER
IN OHIO call (216) 289-7500 (COLLECT)

The perfect supplement for your NEWDOS, from IJG, Inc.

"TRS-80™ DISK AND OTHER MYSTERIES"

by Harvard C. Pennington

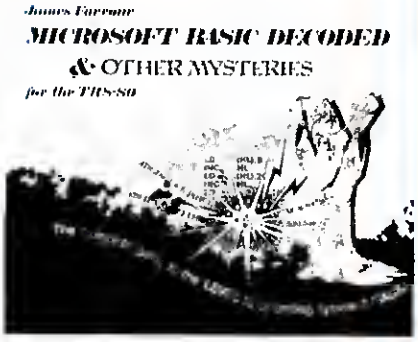
132 pages written in PLAIN ENGLISH packed with HOW TO information with details, examples and in-depth explanations. Recover lost files and directories, remove file protection, make BASIC programs unlistable. How to use SUPERZAP, recover from DOS errors and MORE!

TRS-80™ DISK $19.95

"OTHER MYSTERIES" VOLUME II

by

James Farvour



Call now and place your order for this new book, "MICROSOFT™ BASIC DECODED & OTHER MYSTERIES for the TRS-80™", from IJG, Inc. A primer for cassette and disk BASIC on the TRS-80™, the information provided applies to similar MICROSOFT™ BASIC interpreters. Features include definition of terms, an overview of BASIC and DOS, explanation of exits, error codes, verb actions, "cold" and "warm" restart procedures and examination of system utilities, arithmetic support and I/O driver routines, and the communications region in RAM. Individual routines are explained in detail, with an index provided for easy access. Appendixes include tables for BASIC and DOS vectors, stacks and interrupt locations, PLUS thousands of comment lines for the complete MICROSOFT™ BASIC.

MICROSOFT™ BASIC DECODED . . $29.95

Let your TRS-80™ Teach You

ASSEMBLY LANGUAGE

REMSOFT's unique package, "INTRODUCTION TO TRS-80® ASSEMBLY PROGRAMMING" includes ten 45-minute lessons on audio cassettes, a display program for each lesson providing illustration & reinforcement, and a text book on TRS-80® Assembly Language Programming. Includes useful routines to access keyboard, video, printer and ROM. Requires 16K - Level II, Model I.

REMASSEM-1 . . . . $69.95

FOR DISK SYSTEMS $74.95

Let Your TRS-80™ Teach You

ASSEMBLY LANGUAGE DISK I/O TECHNIQUES

REMSOFT does it again! REMDISK-1 is a concise, capsulated supplement to REMASSEM-1. Package consists of two 45-minute lessons on audio cassettes, and display programs providing illustration and reinforcement. Provides specific track and sector I/O techniques, and sequential and random file access methods and routines.

REMDISK-1 . . . . $29.95

MICROPARAPHERNALIA

DISKETTES (box of ten)

5¼" PLAIN JANE™ $21.95
5¼" PLAIN JANE™ *Gold* $25.95
5¼" DATALIFE™ MD 525-01 . . $26.95
8" DATALIFE™ FD34-8000 . . . . $43.95

NEWDOS by APPARAT

NEWDOS/80 by Apparat $149.95
NEWDOS + to
NEWDOS/80 UPGRADE . . . . . CALL
NEWDOS + with ALL UTILITIES
35-track $69.95
40-track $79.95

BOOKS

TRS-80™ DISK
AND OTHER MYSTERIES . . $19.95
MICROSOFT™ BASIC DECODED $29.95
1001 THINGS TO DO WITH YOUR
PERSONAL COMPUTER . . . . $ 7.95

1001 THINGS TO DO WITH YOUR PERSONAL COMPUTER

BY MARK SAWUSCH

333 pages $7.95

333 pages, written in simple terms, of "what-to-do" and "how-to-do-it". Suitable not only for microcomputers, but for programmable calculators as well. Includes program listings, formulas, a glossary of computer terms and more! Definitely a MUST BUY!

A PARTIAL LIST OF APPLICATIONS

Real Estate Evaluation	Test Your Typing Speed
Astrology	Finances & Investments
Income Tax	Biorythm
Speed Reading	Energy Efficiency
Personality Test	Antenna Design
Statistical Analysis	Letter Writing
Logic Circuit Analysis	Recipe Index/Calculator
Carpenter and Mechanic's Helper	
General Purpose Clock Timer	

"OTHER MYSTERIES" VOLUME III

by Dennis Kitsz

Call now and place your order for this new book, "THE CUSTOM TRS-80™ & OTHER MYSTERIES", from IJG, Inc. More than 300 pages, with over 60 photographs, of projects for the hardware hobbyist. Includes schematics, PC layouts, software driver code, etc. for such do-it-yourself undertakings as high resolution graphics, reverse video, real-time clock/calender, music synthesis, ROM/RAM additions and more!

THE CUSTOM TRS-80™ $29.00
CALL FOR AVAILABILITY

MORE PRODUCTS

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation. DATALIFE is a trademark of VERBATIM. PLAIN JANE, AIDS-I, AIDS-III, CALCS-III, CALCS-IV, MERGE-III are trademarks of MTC.
© 1981 by Metatechnologies Corporation, Inc.

~20

MOST ORDERS SHIPPED WITHIN ONE BUSINESS DAY

Products damaged in transit will be exchanged.

PRICES IN EFFECT June 1, 1981 THRU June 30, 1981.

Prices, Specifications, and Offerings subject to change without notice.

8106

WE ACCEPT
- VISA
- MASTER CHARGE
- CHECKS
- MONEY ORDERS
- C.O.D.

- Add $3.00 for shipping & handling
- $3.00 EXTRA for C.O.D.
- Ohio residents add 6½% sales tax.

80 AID

DEC Interface, Anyone?

We have a TRS-80 Model 1 Level II with an expansion interface and 64K. We also have a DEC (Digital Equipment Corp.) VT78 with two dual RX01 disk drives.

Wouldn't it be nice if the TRS-80 could utilize some of this disk space!

We would be very interested to hear from anyone who has a bright idea and/or experience at interfacing such equipment either directly or via the processor, which is basically a PDP8.

Stephen and Margrit Walsh
Birkenweg 8
6024 Hildisrieden (LU)
Switzerland

Cassette Woes

I seem to have developed a random problem with my CTR-41 cassette recorder in use with a Model 1, Level II, 32K system.

Occasionally during a CLOAD verification of a CSAVEd tape dump, the check will come up bad. At this point the tape transport stops and a check of the tape shows what appears to be a perpendicular line across the tape. In most cases the tape can be cleaned (erased) using an RS-44-210 bulk eraser, and then CSAVEd and CLOADed all right.

I have experienced this problem several times on various quality tapes and am wondering if this could be unique to my system. I have heard of the CTR-80 recorder having some problem along this line, but do not know whether it is a hardware or ROM routine problem.

Apparently I have the earlier Level II ROM which also has the infamous POKE 16553,255 problem to solve the data read routine from restoring each read.

Any help you might furnish in this regard would be appreciated.

William J. Weaver
714 St. George Dr.
Washington, IN 47501

Pocket Computer Interface

Mr Richmond, February 1981 "Input," is looking for a device that transfers data between the Tandy Pocket Computer and the TRS-80. This interface is available from:

Rainhard Wiesemann
Winchenbachstr. 3a
D-5600 Wuppertal 2
W. Germany
Tel: 0202/514044
TELEX: 8 591 617

The device is accompanied by a machine code program (MTERM32 MTERM48) on disk for disk systems with 32K and 48K. Also included are demo programs and an extensive description.

I have tested both the hardware and software and I am amazed with it. Data and BASIC programs may be transferred to the TRS-80. Programs that will allow downloading the Pocket Computer will be announced in the future.

Gunter Hochstatter
43 Essen 1
Tommesweg 57
Ruf 71 39 36

Pocket Computer Interface

TRSDOS POKE

Under Level II BASIC I know that I can POKE 16396,23 to turn off the break button, and POKE 16396,201 to turn it back on. And under NEWDOS, I can POKE &H5BA5,0 to turn off the break, and POKE &H5BA5,1 to turn it back on. But what about under TRSDOS 2.3? What can I POKE to turn on and off the break button?

Alan Dardik
270 Highwood Ave.
Tenafly, NJ 07670

Connections

I would like to obtain information on connecting a FSK/AFSK unit to use with a Macrotronics M-80 (M-800) Ham Interface (TRS-80) in conjunction with a KENWOOD TS-520. Any type of information is appreciated.

James Gonsalves, Sr.
2257 Manhattan Place
Santa Clara, Calif. 95051

Softball Software

Is anyone aware of any programs for compiling baseball statistics that would run on a TRS-80 Model II (64K, one disk)? As both an avid software player and microcomputer owner, I can attest that such a program would find a ready market. For example, in San Francisco alone, there are over 400 softball teams involving over 7,000 players.

I would be more than happy to work with a skilled programmer (which I am not) in developing and marketing a multi-level program that could provide a range of statistical data to subscribing teams. More specifically, I have developed a conceptual design for this type of program and identified over 60 output measure that describe hitting, fielding, and pitching performance.

Edmund F. Fennessy
1841 24th Avenue
San Francisco, CA 94122

80 REVIEWS

Edited by Pamela Petrakos

by G. Michael Vose

Word processing is a major advancement in the science of communication—a science that has become crucial in the post-industrial world. Sophisticated use of our language is important in every field.

Word processing gives the writer, editor or proofreader the freedom to be creative without being bogged down with the details of producing error-free copy. Anyone who has ever written for an audience has experienced the exhilaration that comes from changing a single word to give a sentence just the right meaning—and most of us have also experienced the frustration of having our well chosen words misunderstood, sometimes because of our own cleverness. Word processing could make us all better writers by simply making rewrites easier, freeing us to evaluate our written efforts more thoroughly and critically.

Word processing for microcomputers is a recent phenomenon, but great strides are being made in this area. We'll take a look at three of the latest TRS-80 software packages in this overview of word processing, circa 1981.

What is Word Processing?

How does one process words? The two major functions of word processing software are: text editing and text formatting. Text editing provides the ability to enter text, and then go back over it to correct mistakes and add or delete words and single characters. A sophisticated text editor will also permit search operations and search/replace operations.

Text formatting provides the ability to set parameters for the printed copy of your text. It allows you to set up your printed page format for line length, lines per page, margins and other standard parameters. The more sophisticated formatters also support text centering, headers and footers, underlining, superscripting, subscripting, bold type and other functions (i.e., specialized symbols).

The most obvious ability that a good editor provides is the ability to correct the most common mistakes we make when writing—spelling errors, ommissions, errors of punctuation and the like. This is most often accomplished by moving a cursor to the area of a mistake, and overtyping or inserting (or deleting) incorrect characters. A sophisticated text editor provides much more. If you had to go back and correct the spelling of a word that appeared many times in the text, you could go blind or crazy, or both. With a global search and replace word command, the computer would do all the drudgery.

Good word processors provide a block move feature that allows the transfer of entire paragraphs, or any size block of text, from one location to another. A block delete permits the deletion of any size chunk of text. The insert mode of a good word processor allows the addition of unlimited amounts of new text, wherever it is needed.

Text formatting is the icing on the cake. Now you've written a top-notch report using the word processor's editor, with sophisticated text formatting you can make that report look like it was produced by a professional printer. The components of good formatting go far beyond mere margin assignment, double spacing or page numbering. They include headers and footers that appear on the top and bottom of every page; proportional spacing to achieve justified left and right margins for the typeset look; tab settings; single keystroke paragraph signals; support for underlining, boldface type, subscripts, superscripts; and support for special characters or type fonts.

Most word processing systems allow command characters to be embedded in the text to produce the format functions desired. This is how the most elaborate features are called by the system.

Since there is a wide variety of printers available for use with micro and minicomputer systems, the major weakness of most word processing systems is an inability to support more than one printer. If, after all, you have set up the print format to look as attractive and professional as possible, then give the print command and the system just sits there doing nothing. . . you know you have a problem, particularly if you forgot to save the text before the system hung up. It is important that the user determine which printers his software will support *before* using that package.

Scripsit and Electric Pencil have been around for some time now; there have also been some inexpensive entrants, such as The Wordslinger and PensaWrite1, into the word processing derby. The three most recent word processors on the market are: **LazyWriter** by **ABC Sales for $125, Pensa-Write2** by **Pensadyne Computer Services** for **$79.95**, and **Subedit/Subscript** by **ProSoft Software** for **$39.95** (with enhancements: $59.90). Let's take a look at some of the major features of these new offerings.

Fig. 1 will provide a quick overview of the major features of each package. Both LazyWriter and Pensa-Write2 are written in machine language and offer sophisticated text entry and editing features. The Subedit/Subscript package is written in BASIC. All three are configured for a TRS-80 Model I with at least 32K of memory and a disk drive. More memory usually allows you to process larger files.

The producers of this software plan enhancements to their individual packages, and all three offer some kind of plan to allow purchasers to acquire updates at nominal cost. This usually involves registration of the purchase of the software with the manufacturer, and will require the return of the original disk in order to obtain an upgrade. All three provide thorough documentation (usually produced with their own software!) and a place to write or call if you have trouble.

Each software package offers something unique. **LazyWriter** has a Model III version on the drawing board. It also features a unique cancel edit feature which allows you to cancel an editing change should you decide you really don't want to make it. LazyWriter also allows loading of Electric Pencil files as well as ASCII saved files and ASCII saved BASIC programs. There is a Help command which explains the features most often used, and there are ten user-definable command keys that can be programmed for special functions.

Pensa-Write2 has an intriguing module support feature. It allows you to append program modules (available from Pensadyne) such as mailing list, financial report generator or special printer support modules to the main program. Or, you can write your own machine language modules and use any one of up to ten commands that you create to call them up. The program also keeps track of free disk space and allows inspection of the files in the directory without an exit to DOS.

The **Subedit/Subscript** program is modeled after the CMS Editor used on the IBM System/370 mainframe computer. The user's manual claims that familiarity with this Editor makes use of the manual unnecessary! Even though Subedit/Subscript is written in BASIC, it runs very fast. Its run time has been increased by a ProSoft utility called Faster. Of all word processors for the TRS-80, this program's text formatter has the most features. Of course, any formatter is limited by what its printer can produce.

Compare and Contrast

Nothing in this world is perfect and these three packages have some minor problems. All perform the task at hand admirably but all also have their little annoyances.

LazyWriter is the easiest of the three packages to use. Its text editing commands are called with single keystrokes (I for insert, O for overwrite, etc.) and its screen displays are simple and uncluttered. When you are in a text editing mode, that mode's name is displayed at the bottom of the screen so that you won't forget where you are. To exit that mode, you merely press (enter). Its major drawbacks are in the print formatting area. As Fig. 1 shows, it cannot support headers, footers, underlining, page numbering and the like. There is a character counter at the bottom of the screen during text entry that lets you know where you are in relation to the end of the current file. It has a built-in lowercase driver and is fast enough to stay ahead of the nimble fingered typist.

The Pensa-Write2 package has substantially more complicated screen displays and system commands. It has excellent text formatting capabilities which are easy to modify. The system does not provide a lowercase driver but will support the Radio Shack driver at the expense of the on-board clock. There are no flaws in this package and the documentation provides excellent flowcharts to help you decipher the intricacies of its command structure.

X = Yes
— = No

	Lazy-Writer	Pensa-Write2	Subedit/ Subscript
Insert Text	X	X	X
Page Scrolling	X	X	X
Block Move	X	X	X
Lowercase	X	—	—
Cancel Edit	X	—	—
Merge Files	X	X	X
Wraparound	X	X	X
Justify Text	X	X	X
Margin Formatting	X	X	X
Subscripts	—	—	X
Underlining	—	—	X
Overstrike Bold	X	—	X
Page Numbering	—	X	X
Headers/Footer	—	X	X
Price($)	125	80	40

Fig. 1. Word Processors Compared

Subedit/Subscript has some excellent features and has great potential. Unfortunately, it also has a couple of serious problems. The text formatter, Subscript, is just about the best formatter now available for the TRS-80. It supports all major printers and has very sophisticated features. Using Epson's MX-80 printer, I got results that were amazing. (If you haven't seen the MX-80 do its tricks, put that on your list of things to do.) Subscript supports the MX-80's overstrike, emphasized and double emphasized and compressed printing. With its support of subscripting, superscripting, underlining, bold print and other features, Subscript really shines.

Subedit, on the other hand, features a text entry problem that bothered me. Because it is written in BASIC, it stores text in 255 character strings. This makes it necessary to press (enter) at the end of every line of text. Thus, typing is like using an ordinary typewriter, except that there is no little bell to warn you of the approach of the end of a line. For an additional $19.95, you can buy ProSoft's Mininit utility. This utility will provide a lowercase driver, repeating key functions and a few other goodies. The system will also work with Radio Shack's lowercase driver if you want to avoid that expense.

The Bottom Line

There is no question that these new word processors and the others already on the market combine to make the $10,000 to $20,000 investment in a big name, dedicated word processor a luxury rather than a necessity.

Which software package is the best? For value, if you don't mind sacrificing a few features, the ProSoft entry is the best bet at $39.95. Lazywriter is by far the easiest of the three to use and, with enhancements to its text formatter, will someday be worth its price. The classiest package in this group is the Pensa-Write2 package —the price is moderate, the features are solid and professional and it will be worth the extra time you spend learning its complexities.

Of course, there are other word processor software packages on the market and there are probably more being written. Each new generation of these packages will refine and enhance what has gone before. The result can only be a boon to writers everywhere.■

Ed. Note: Pensadyne has just completed a revised version of Pensa-Write 2 which they are calling Pensa-Write 2.1. Present owners of Pensa-Write 2 are eligible to receive the updated version. Contact Pensadyne for specific information.

The Micromatic-80 Printer
Micromatic Corp.
Indianapolis, IN
$795

by David E. Clapp

The Micromatic-80 is an excellent output device for the TRS-80. It consists of a used, heavy duty IBM 1980 terminal system and an interface device. The interface is enclosed in a small cabinet about the size of a dictionary.

The typewriters are not new, but they are reconditioned and tested by the Micromatic Corporation before shipment. The typewriter and the interface are interconnected by a short cable. The interface, in turn, is connected by ribbon cable to the TRS-80 output port, either directly or through the expansion interface.

The typewriter arrives equipped with a BCD ball which can be supplemented by purchasing other IBM selectric typing balls (10 pitch only). This interchangeability is a valuable feature, permitting an interesting variety of output formats (script, letter gothic, prestige elite, etc.). The interface has a code switch so the user can select BCD or correspondence code.

Whenever using a standard IBM typewriter ball, the switch must be in the correspondence position.

The Micromatic-80 system can be used off-line as an ordinary office typewriter. In this mode, the typewriter must be operated with the BCD ball only. Standard line printers are typically only usable when connected to the computer.

The major advantage of this system is letter quality output. This feature is important for generating output equivalent in appearance to common business correspondence. It is impossible to distinguish the output of the Micromatic-80 from original typewritten copy. For applications demanding originally typed, letter quality copy—the Micromatic-80 is especially effective.

"The Micromatic Corporation stands behind their product. . . . They spared no cost ensuring that I had a working, satisfactory system."

Disadvantages

The system has certain disadvantages. Rate of output is especially slow; eight to nine characters per second. While this rate could be increased, the manufacturer feels the present rate is most appropriate for used equipment. Knowing that the copy will require five to ten minutes for output allows the user to step away from the computer and engage in other activities while the output is completed. Typically, most text is composed, edited, and stored on some peripheral device, then left alone while copy is generated.

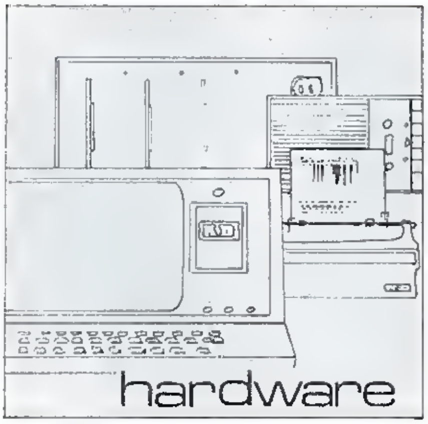
Another potential disadvantage is possible service or repair requirements. An IBM selectric is a complex device which will require periodic maintenance and adjustment. Certain maintenance can be performed by the user, but other tasks may require a trained technician. Most typewriter shops will service IBM selectrics; however, since these typewriters are terminals some shops may refuse to work on them or charge more. A final alternative is to obtain service directly from IBM, which is probably the most expensive option.

The Micromatic-80 works beautifully with Radio Shack's Scripsit. A lowercase modification is required to properly prepare correspondence. The least expensive modifications will work for the Micromatic-80 system (typically $19.95); these modifications do not display descenders on the screen. Most lowercase modifications require a short driver routine to reverse the keyboard (shift for uppercase like an ordinary typewriter). Fortunately, this driver is not required with Scripsit and the Micromatic-80 system.

When operating Scripsit and the Micromatic-80, the operating procedure is quite simple. The first step is to power up the system, load Scripsit, and begin typing! No other software is required. The lowercase option, keyboard debounce and keyboard reverse are all operational.

The Micromatic-80 requires a specific power-up sequence: turn on typewriter, turn on interface, turn on TRS-80. This protects the TRS-80 from incoming spikes from the Micromatic-80 system. If the sequence is ignored, the spikes will frequently bomb the resident program as well as threaten the electronics of the TRS-80.

The selectric typewriter is noisy both at idle and in operation. Typically, the user may wish to leave the Micromatic-80 system off while preparing copy. After the copy is edited, formatted, and ready for output, the Micromatic-80 can be activated. However, this sequence requires computer shut-down which, in turn, re-



quires reloading of the program and text copy. Reloading is no problem with high speed peripherals, but if cassette tape is used, considerable time is required to reload Scripsit and the text copy. One could avoid this delay by simply allowing the Micromatic-80 system to idle while preparing copy and enduring the low background rumble of the typewriter motor. Enclosures are available (or can readily be built) which will reduce this noise to an acceptable level.

The Micromatic-80 system will accept either fanfold or ordinary single sheets of paper. A tractor feed option is available, but the friction feed option will feed fanfolded paper satisfactorily. When feeding individual sheets, the user can use the "print, pause" feature of Scripsit which delays printing until the next sheet is input.

The Micromatic-80 system does not have a tabulation function. The user will find that extensive tabulation will be time-consuming since the typewriter must individually count the spaces to a tab location.

The Micromatic-80 system represents a quality investment for quality output for the TRS-80. The selling price is competitive with any printer on the market. It is an excellent means to economically produce letter quality output. Many features do not compare to the daisy wheel printers on the market, but the appearance of the output is equivalent. The use of the system off line is valuable feature which is unavailable with a dot matrix or a daisy wheel printer.

The Micromatic Corporation stands behind their product. I had some initial difficulty with my system, and they stayed with me until it was rectified. They spared no cost ensuring that I had a working, satisfactory system. This system is an excellent buy, and I recommend it for anyone seeking a moderately priced letter quality printer. ■

META TECHNOLOGIES

26111 Brush Avenue, Euclid Ohio 44132
CALL TOLL FREE 1-800-321-3552 TO ORDER
IN OHIO, call (216) 289-7500 (COLLECT)

MTC AIDS-III™

MODEL I . . .$69.95 **MODEL II . . .$99.95**

Introducing the latest addition to MTC's family of data management systems, AIDS-III. NO PROGRAMMING, easy to use. COMPLETE PACKAGE including demonstration application, documentation and MAPS-III (see below).

- Up to 20 USER-DEFINED FIELDS of either numeric- or character-type.
- CHARACTER-type fields may be any length (total: up to 254 characters).
- NUMERIC-type fields feature automatic formatting, rounding, decimal alignment and validation.
- Full feature EDITING when adding or changing records:
  - ENTER FIELD (can't type-in more characters than specified).
  - BACKSPACE (delete last character typed).
  - DELETE FIELD contents.
  - RESTORE FIELD contents.
  - RIGHT-JUSTIFY FIELD contents.
  - SKIP FIELD (to next or previous field).
  - SKIP RECORD (to next or previous record).
- SORTING of records is MACHINE CODE assisted.
  - 200 RECORDS (40 characters) in about 5 SECONDS.
  - ANY COMBINATION of fields (including numerics) with each field in ascending or descending order.
- SELECTION of records for Loading, Updating, Deleting, Printing and Saving is MACHINE CODE assisted.
  - Specify up to 4 CRITERIA, each using one of 6 RELATIONAL COMPARISONS.
  - LOAD or SAVE selected records using MULTIPLE FILES.
  - Example: Select records representing those people who live in the state of Colorado, but not in the city of Denver, whose last names begin with "F" and whose incomes exceed $9000.00.
  - Example: Select records representing those sales made to XYZ COMPANY that exceed $25.00, between the dates 03/15 and 04/10.

MAPS-III (MTC AIDS PRINT SUBSYSTEM), included at no charge, has the following features:

- Full AIDS-III SELECTION capabilities.
- Prints user-specified fields DOWN THE PAGE.
- Prints user-specified fields in titled, columnar REPORT FORMAT, automatically generating column headings, paging and (optionally) indentation.
- Can create a single report from MULTIPLE FILES.
- Prints user-defined formats for CUSTOM LABELS, custom forms, etc.

BELOW ARE TESTIMONIALS from owners of AIDS systems. These are absolutely authentic statements and are typical of the comments we receive.

"This program will do more for my business than all the other programs I have, combined."
David Wareham, Vice President (EDP), National Hospital and Health Care Services Inc.

"We have 32 different Data Base Management packages for the TRS-80. AIDS-III is easily the best. It also makes it easier for us to step up to our Model II since the package is available for both computers."
Jack Bilinski, President, 80 Microcomputer Services

"Your AIDS program is far and away the finest information management system that I've ever seen. I am currently using it to maintain a clear picture of the demographic data on all the kids in our residential treatment program and it is working for me superbly."
Frank Boehm, Director, Front Door Residential Treatment Program

MTC AIDS CALCULATION SUBSYSTEM-III™

MODEL I . . .$24.95 **MODEL II . . .$39.95**

MTC's most popular AIDS subsystem. Use for report generation involving basic manipulation of numeric data. Features are:

- User-specified page title
- Columnar Headings
- Optional Indentation
- Use for accounting, inventory, financial and other numeric-based information systems.
- Columnar subtotals generated when there is a change in a user-specified column.
- User-specified Columnar Totals
- Columnar values computed using constants and/or column values
- Balance forward calculations (Ex: Gross sales equals previous gross sales + sale amount + sales tax).

Compare AIDS-III™/CALCS-III™ with any other data management package under $100!

CALCS-III™ REQUIRES THE PURCHASE OF AIDS-III™

AIDS OWNERS! WE HAVE WHAT YOU'VE BEEN WAITING IV. . .

MTC CALCS-IV™, that is.

- More Computations
- Save Report Formats on Disk
- Faster, and more!

MTC CALCS-IV™. $39.95
For Model II. $59.95

CALL FOR CALCS-III UPGRADE PRICING

MTC AIDS MERGE-III™

This subsystem will combine up to 14 AIDS-created data files into a single, large file. An optional purge capability removes duplicate entries while performing the merge operation (can even be used to eliminate duplicates in a single file). Machine-code assisted for high-speed performance, MERGE-III™ properly handles files sorted by any combination of fields, including numerics, with each field in ascending or descending order.

MTC AIDS MERGE-III™. $19.95
For Model II $29.95

MICROPARAPHERNALIA

DISKETTES (box of ten)

5¼" PLAIN JANE™. $21.95
5¼" PLAIN JANE™ *Gold* $25.95
5¼" DATALIFE™ MD 525-01 . .$26.95
8" DATALIFE™ FD34-8000 . . . .$43.95

NEWDOS by APPARAT

NEWDOS/80 by Apparat. $149.95
NEWDOS+ to NEWDOS/80 UPGRADE . . . . .CALL
NEWDOS+ with ALL UTILITIES
35-track $69.95
40-track $79.95

BOOKS

TRS-80™ DISK AND OTHER MYSTERIES . .$19.95
MICROSOFT™ BASIC DECODED $29.95
1001 THINGS TO DO WITH YOUR PERSONAL COMPUTER . . . .$ 7.95

MORE PRODUCTS

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation. DATALIFE is a trademark of VERBATIM. PLAIN JANE, AIDS-I, AIDS-III, CALCS-III, CALCS-IV, MERGE-III are trademarks of MTC.
© 1981 by Metatechnologies Corporation, Inc.

✓20

MOST ORDERS SHIPPED WITHIN ONE BUSINESS DAY

Products damaged in transit will be exchanged.

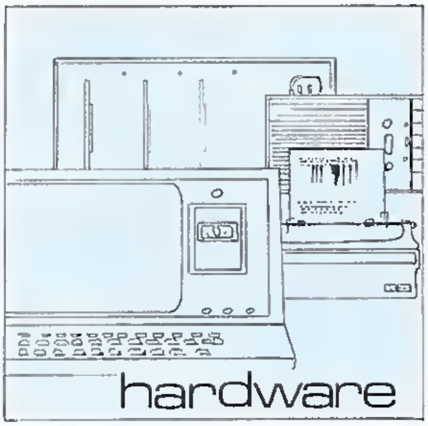
PRICES IN EFFECT June 1, 1981 THRU June 30, 1981.

Prices, Specifications, and Offerings subject to change without notice.

8106

WE ACCEPT
- VISA
- MASTER CHARGE
- CHECKS
- MONEY ORDERS
- C.O.D.

- Add $3.00 for shipping & handling
- $3.00 EXTRA for C.O.D.
- Ohio residents add 6½% sales tax.



Radio Shack Line Printer VI
Tandy/Radio Shack
Fort Worth, TX
$1160

by Richard C. McGervey

If you've been looking for a line printer, you know what a jungle the business can be. Prices range from a couple hundred to a few thousand dollars, and features and functions of the printers are just as varied. Expensive printers are usually far beyond the resources of the hobbyist. On the other hand, low cost printers are slow, limited in function, and likely to give your computer system a case of heartburn that will reduce it to a smoldering heap of electronics on your desk top. Unless you want to pay a good price you are not going to get a good printer.

What are the alternatives? You can do without, or you can risk buying a cheap printer and pray it works without vaporizing your computer. You can go all out and buy a $3000 word processing printer. Finally, you could get Radio Shack's new Line Printer VI. For the money, I believe it is the best choice.

I had a Radio Shack Quick Printer II, which was fine for numerical data runs and was cheap and reliable, but it wasn't a line printer. I finally decided to spend the money and get a good line printer. I wanted a lot of features: tractor and friction feed, graphics, a print font that would be acceptable for word processing, a paper out warning, bi-directionality, adjustable width that would handle everything from labels to 15 inch computer paper, and a good print rate. I didn't want to pay more than a thousand dollars.

I took my request for a printer with my list of requirements to a friend of mine who happens to be a Radio Shack manager. All he had was an advertisement for a new line of printers, including the Line Printer VI. It sounded good, but delivery time had not even been set. I put in an order just so I could see what the printer looked like, and left without much hope.

A few days later I received a call from my friend. He had managed to get one of the first printers released, and I could have it if I wanted it. I was desperate so I decided to take the printer and hope it was what I wanted. It turned out to be much more than I expected. Well designed and fast, it had all the features I wanted and more.

Special Features

I wanted a tractor feed that was adjustable from 2 1/2 to 15 inches. I also wanted single sheet friction feed. The Line Printer VI had that and the added feature of a removable tractor. The tractor, which is almost flush with the top of the printer, snaps in and out.

The Line Printer VI has a paper out alert that stops the printer without losing data when the paper runs out. This means I can leave the printer unattended while it prints long data runs without worrying about returning to find my carriage receiving a nice coat of ink. The paper out warning works whether the paper is fed from the bottom or from the rear of the printer. After refilling the paper and resetting the printer, it takes up where it left off with no data lost.

Dimensions, Type and More

The Line Printer VI is not small; at 24.2 inches wide by 6.3 inches high by 13.3 inches deep, it does require some room. A separate, sturdy desk is recommended. Since the tractor is not a large superstructure, the weight and height are not prohibitive. Remember that a 15 inch wide carriage requires a fairly large printer.

The printer fonts available (four) are shown in Fig. 1. Power up mode is 132 characters per line (at 15 inch paper size). This will print at 100 characters per second and 33 lines per minute. The normal characters may be elongated to double width, or compressed to 120 characters per second, 37 lines per minute. The compressed mode can also be elongated into the compressed—elongated mode. In either elongated font, the bi-directionality of the printer doesn't work. It does func-

```
POWER UP MODE - 132 CHAR.
E L O N G A T E D   M O D E
COMPRESSED MODE
COMPRESSED - ELONGATED MODE
```

Fig. 3

```
THIS IS 6 LINES/INCH
THIS IS 6 LINES/INCH
THIS IS 6 LINES/INCH
THIS IS 8 LINES/INCH
THIS IS 8 LINES/INCH
THIS IS 8 LINES/INCH
AND 12 LINES/INCH
AND 12 LINES/INCH
```

Fig. 2

```
 !"#$%&'()*+,-./Ø123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]↑←
`abcdefghijklmnopqrstuvwxyz{¦}~
 àç£ µ°▾†§ ©¼ ½‖¥ÄÖÜ¢ äöüß èùé¨ƒ
 ▘ ▝ ▖ ▗ ▚ ▞ ▀ ▄ ▌ ▐ ▛ ▜ ▙ ▟ █
┌ ─ ┐ ┬ ├ │ └ ┘ ┴ ┤ ┼ ◤ ◢ ◥ ◣
```

Fig. 1

tion fully in either compressed or normal modes.

I was especially impressed that once a particular font is selected it will remain active, unlike many printers on which you must call special fonts after each carriage return. Also, when a print font is cancelled, the printer returns to the font that was active previously, not necessarily the power up mode.

Line spacing, called pitch, is another important consideration. The Line Printer VI powers up at six lines per inch. There is also an eight lines per inch mode that is software selectable. Finally, the 12 lines per inch pitch is available for graphics. The pitches remain active until cancelled. See Fig. 2 for an example of the three available line pitches.

Fig. 3 shows the complete character set, including graphics and special symbols. The characters in Fig. 3 were printed in the normal mode for clarity. In the elongated modes some of the special symbols are not clear. The printer has a full upper/lowercase. Like most printers, it prints a bracket for an up arrow.

The print is satisfactory for word processing use; it is not as fancy as an impact printer, but the letters are neat and clean.

Most printers available to hobbyists are slow. Forty characters per second is not uncommon. In the normal mode, the Line Printer VI will print 100 characters per second.

If you watch a unidirectional printer, you will notice that the character per second rate is clocked only while the print head is active. The time of the carriage return and the time of inactive print head travel is not added in. Also, most printers make a full left to right travel regardless of where they print on the page. Eighty characters per second is rather slow when all this unused time is added in.

The Line Printer VI is not plagued by any of this slowness. Since it is bidirectional, it prints on the carriage return as well as left to right travel. Also, when printing tabbed data, the print head moves to the tabbed position the first time and then returns to the tabbed start position only as long as more tabbed data is available. This means that no time is wasted returning full left and then to the tabbed position with each line. The same is true at the other end of travel; the printer will print only until the line end and then returns. The only exception to this is when LLISTing a BASIC program. If the line is 200 characters long and you have nine inch paper, the printer will continue off the paper and onto the roller.

Everything considered, the Line Printer VI is faster than other printers that claim the same or higher character per second rates. Speed may not be a requirement for you but you have to admit that it would be a welcome extra. For speed and economy, the Line Printer VI can't be beat. ■

MT-32 Printer/Interface Module
Microtek Inc.
San Diego, CA
$119.50–$199.50

by Fritz Milhaupt

Expansion of the TRS-80 beyond the Level II, 16K limits of the keyboard/CPU unit has always presented the question of which expansion system is best suited for your needs. With expansion interfaces available from Radio Shack, Lobo, Exatron, Microtek and an expansion board from LNW research, it is difficult to choose the most economically designed and priced unit.

Until Microtek introduced their MT-32 Printer/Memory Module, there was no way of adding just additional memory and a parallel printer interface to your system short of building one yourself. The MT-32 provides both of these features for less than $125 in its most basic form.

This peripheral is considerably cheaper than Radio Shack's expansion interface. It only costs $119.50 for a unit with no additional RAM. 32K (MT-32B) and 48K (MT-32C) models are also available for $159.50 and $199.50 respectively. The MT-32, like Radio Shack's interface, sits under the video monitor so it takes up no additional desk space.

Documentation

The documentation and instructions are clear and straightforward; however, there is a difference in the memory installation instructions. The instructions included with the interface stated that the first additional set of 16K chips should be installed in sockets U13-U20 of the unit. The instructions in the advertising brochure stated that the chips were to be installed in sockets U5-U12. After some trial and error testing, I found that the correct sockets were U5-U12, as stated in the brochure.

Memory installation couldn't have been made any easier. All that is required is to remove two screws on the back of the module, slide the cover off, insert the chips, slide the cover back on and replace the screws.

"I have experienced no problems with it (the MT-32) in the three months that I have been using it."

The only disadvantage that I could find was that the MT-32 has no extension of the CPU's bus for connection to other peripherals. Fortunately this problem is easily solved by the installation of any of the commercially available "2 for 1" bus splitting cables between the MT-32 and the keyboard/CPU unit.

Although I am presently unable to test the printer interface portion of the expansion module (until I can scrape enough together for a good printer), I have been assured through calls to Microtek and Radio Shack's computer services department that any cable used for interfacing a printer to the Radio Shack expansion interface will work with the MT-32 as well.

One of the major advantages of this interface is that since it has no disk controller, pressing the reset button to stop the cassette recorder or the printer will not result in a hang-up or return to Memory Size? but will return to Ready as it would without the interface. Another great advantage is that the interface draws its power from the keyboard's power supply so that another cord needn't be added to the already impossible tangle of cords behind the computer.

The power supply is simply plugged into the back of the interface. A cord from the MT-32 to the keyboard is connected to the keyboard power jack to feed the CPU. All power-ups are now handled by pressing the button on the front of the module.

In summary, the MT-32 is a great, low cost alternative to the $300 plus price of the Radio Shack expansion interface. I have experienced no problems with it in the three months that I have been using it.

Even if you don't want a printer but need the extra memory, the MT-32 is the lowest cost product for memory expansion. The money saved by using this interface can be used for other worthy causes such as the purchase of a printer or other peripheral. ■

General Accounting Package
Microed
San Diego, CA
$440

by Helen Huffman

Before the TRS-80 Model II, I had an Imsai with CP/M as an operating system. I purchased a Model II as an upgrade and because it could better serve our business. I basically wanted to use it to computerize the company accounts. Because I liked using a CP/M operating system, I considered getting it for my new Model II. I was also shopping around for an accounting package, and was pleasantly surprised when I received a product announcement from a company in California called Microed, offering both the CP/M system and an accounting package for the Model II. It appeared to be what I needed.

I received the software package, which contained manuals and two disks. The first disk contained all the CP/M programs and 11 general ledger programs. The 11 ledger programs alone occupied 218K bytes of space, in executable code, not BASIC. (It probably wouldn't fit on the disk if it were written in BASIC.)

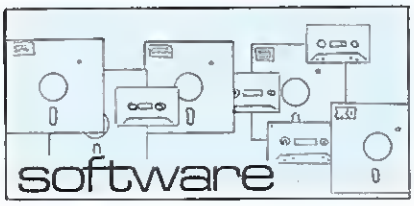
Documentation

The first Microed manual was a summary of CP/M programs and a description of Microed's written programs for CP/M. This manual provided a background adequate for understanding the use of the programs.

The accounting manual is written in a self-teaching style. It is intended to be used as a step-by-step guide in setting up the accounting system on the computer. It uses a sample list of accounts to explain all the functions of the software. After spending about a half an hour reading through the manual, with not much luck absorbing the material, I decided to do as the manual suggested. I created a data disk and used it to exercise the programs.

I made back-up copies of the disks I received, using Microed's CP/M program to format the disk. I also formatted two extra disks (a data disk and a standby) as suggested in the accounting manual.

The data disk is used in the second disk drive and contains all the accounting program results. This disk also receives all the newly created files. It is similar to the data base concept where the data disk becomes the data base. Following the manual, I used one of the general ledger programs to enter a chart of accounts listed in the manual as good learning examples. The manual slowly guided me through the use of the program.



Step by Step

I spent most of the day going through the accounting manual following the step by step procedures. I printed out reports along the way, when indicated by the manual. Samples are included for comparison. The amount of software was overwhelming. When finished, I felt knowledgeable, yet confused. It was almost too much, too fast. The next day, after reviewing what I had done and re-reading the manual, I felt better about the project I had undertaken. I finally entered some January transactions with ease.

Numbered and Named

I found several suggestions in the manual quite useful. The first was to post a sheet nearby containing a list of the program names and numbers. (The sheet was furnished with the manual.) The programs are "named" with a number which is acceptable under CP/M. Until I am familiar with each program, the list is handy and saves me from referring back to the manual. All the general ledger programs are numbered (named) in the 100s, accounts receivable in the 200s and accounts payable in the 300s. A second suggestion was to post the chart of accounts nearby. When transactions are made, it is handy to be able to quickly scan your chart of accounts for the account number.

When I first found out that the program was written in FORTRAN and I would not have access to the source code, I was somewhat disappointed—I would have no way to modify it for my own situation. I see now that the capability of this accounting package far exceeds my needs, and it may be quite some time before I need to modify it, if ever.

Transactions are entered using program 103, Add General Journal Transactions. The screen printout guides you through the input, and the manual explains the process. Another program, 104, posts transactions and gives you a report containing those transactions. Two additional programs give you a general ledger detail report and a general ledger summary. Program 109 finalizes the month. Program 107, Print a Balance Sheet, and Program 108, Print a Profit and Loss Statement, are for annual summaries or for any update on the financial status of the business.

> *"I have become more and more convinced . . . it is one of the best bargains available . . ."*

Program 110 defines parameters (account numbers, dates) so that you can search out specific information from the data disk. The output is a report under any title you give it, containing the information.

The accounts receivable and accounts payable programs are on the second supplied diskette. Accounts receivable has 10 programs and accounts payable has 11. The basic function of both sets is keeping track of invoices and statements. Accounts payable has the capability to print checks onto a pre-formatted check blank which goes into the printer. The data disk used in the general ledger is also used by both programs, and there are no problems sharing the data base among the different programs.

The CP/M from Digital Research is the 1.4 version. However, it is more than adequate. Microed has written the portion for the Model II, so that either single or double density disks may be used. The densities may be mixed and the system is able to detect the difference. For the average business person, the supplied CP/M programs would not be used. Only the disk formatting and disk copying programs are really necessary for the accounting programs.

I received a bonus in my package. My package came with what Microed called security programs. These programs allow me to use a password for entry into the system, and hence the accounting information. I created my own password and was not allowed to get to the CP/M system until I entered the correct password. It will accept uppercase or lowercase, or any numbers or symbols in any combination, up to eight characters. When I was done, I was able to kill the password feature by simply entering a carriage return when running the program Newpass.

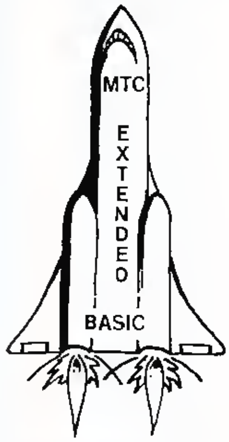
After using this package for several weeks, I have become more and more convinced that it is one of the best bargins available in off-the-shelf software.

In summary, I am quite satisfied with the package. It is successful, in my eyes, because it is a complete package; system software and application software combined to run together. ■

META TECHNOLOGIES

25111 Brush Avenue, Euclid Ohio 44132
CALL TOLL FREE 1-800-321-3552 TO ORDER
IN OHIO, call (216) 289-7500 (COLLECT)

TAKE YOUR MODEL II TO NEW HEIGHTS



MTC is proud to announce MTC EXTENDED BASIC for the Model II, by R. Ryen. Features include "fixes" to existing BASIC, multi-line functions, extending an existing sequential file, PEEK, POKE, greatly enhanced screen control and expanded editing capabilities. The contents of variables are NOT CHANGED when editing, deleting, inserting or merging lines, allowing continued program execution! All this and much more.

MTC EXTENDED BASIC .... $ 99.95

80 – 20 = 129.95

THE ORIGINAL NEWDOS/80

WITH MTC QUE CARD

★ SPECIAL PRICE ★

SAVE $20 **$129.95**

Apparat's long-awaited successor to NEWDOS+ is here! This is not an enhanced version of NEWDOS, but a completely new product. Simplified DOS commands can be instantly executed from BASIC, even within a program, without disturbing the resident code. System options, such as password protection, number and type of disk drives, BREAK key enable/disable and lowercase modification recognition, can be quickly and easily changed. Five new random-access file types allow record lengths of up to 4096 bytes, and no FIELDing! A powerful CHAIN facility allows keyboard INPUTs to be read from a disk file. An improved RENUMBER facility permits groups of statements to be relocated within program code. Diskettes may even be designated as RUN-ONLY! Features all NEWDOS+ utilities (SUPERZAP 3.0, etc.) and much more! One MTC technical staff member said having NEWDOS/80 is "better than sex" (you'll have to judge for yourself!). Includes 180-page instruction manual and MTC QUE card.

NEWDOS/80 # SPECIAL # $129.95
MTC QUE Card only .... $ 7.50

CALL REGARDING OUR NEWDOS+ UPGRADE PRICING

MODEL III VERSION

Has all the features of the Model I version plus enhancements. Allows any mix of single- or dual-sided 40- or 80-track disk drives. Most BASIC and many machine code programs written for the Model I will run without modification. Includes a utility for converting Model I single density to Model III double density.

NEWDOS/80 for Model III $149.95

CALL FOR AVAILABILITY

MICROPARAPHERNALIA

DISKETTES (box of ten)

5¼" PLAIN JANE™ $21.95
5¼" PLAIN JANE™ *Gold* $25.95
5¼" DATALIFE™ MD 525-01 .. $26.95
8" DATALIFE™ FD34-8000 .... $43.95

NEWDOS by APPARAT

NEWDOS/80 by Apparat $149.95
NEWDOS+ to NEWDOS/80 UPGRADE ..... CALL
NEWDOS+ with ALL UTILITIES
35-track $69.95
40-track $79.95

BOOKS

TRS-80™ DISK AND OTHER MYSTERIES .. $19.95
MICROSOFT™ BASIC DECODED $29.95
1001 THINGS TO DO WITH YOUR PERSONAL COMPUTER .... $ 7.95

Michael Shrayer's

ELECTRIC PENCIL VERSION II

for
Model I and Model III

An expanded version of the critically acclaimed original word processing system! Includes all features of Version I plus many new extensions. Runs under most disk operating systems, has improved video text handling, loads any ASCII file for editing (including BASIC files), single sheet mode for printing on letterhead and more! Simple to use, features 2-key commands. An incredible package at an incredible price!

SPECIFY MODEL I OR III

Disk Version $79.95
Tape Version $69.95

CALL FOR AVAILABILITY

Complete for Model I with all utilities
Plus exclusive MTC QUE card!

NEWDOS +

$69.95 by Apparat

Includes REF, RENUM, SUPERZAP, EDITOR/ASSEM., DISASSEM., DIRCHECK, and more! This is the original NEWDOS with all of Apparat's utility programs. Includes exclusive MTC QUE (Quick User Education) card.

40-Track Version $79.95
MTC QUE Card only .... $ 1.50

MORE PRODUCTS

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation. DATALIFE is a trademark of VERBATIM. PLAIN JANE, AIDS-I, AIDS-III, CALCS-III, CALCS-IV, MERGE-III are trademarks of MTC.

✓20

MOST ORDERS SHIPPED WITHIN ONE BUSINESS DAY

Products damaged in transit will be exchanged.

PRICES IN EFFECT June 1, 1981 THRU June 30, 1981.

Prices, Specifications, and Offerings subject to change without notice.

8106

WE ACCEPT
- VISA
- MASTER CHARGE
- CHECKS
- MONEY ORDERS
- C.O.D.

- Add $3.00 for shipping & handling
- $3.00 EXTRA for C.O.D.
- Ohio residents add 6½% sales tax.

Yes, You Can Increase Your Programming Productivity

SNAPP II EXTENDED BASIC

SNAPP II EXTENDED BASIC A family of enhancements to the Model II BASIC interpreter. Part of the package originated with the best of APPARAT, INC.'s thoughts in implementing NEWDOS BASIC. The system is written entirely in machine language for SUPER FAST execution. The extensions are fully integrated into Model II BASIC, and require NO user Memory, and NO user disk space. The package is made up of the following six modules.

XBASIC—Six single key stroke commands to list the first, last, previous, next or current program line, or to edit the current line. Includes quick way to recover BASIC program following a NEW or system or accidental re-boot. Ten single character abbreviations for frequently used commands: AUTO, CLS, DELETE, EDIT, KILL, LIST, MERGE, NEW, LLIST, and SYSTEM.

XREF—A powerful cross-reference facility with output to display and/or printer. Trace a variable through the code. Determine easily if a variable is in use.

XDUMP—Permits the programmer to display and/or print the value of any or all program variables. Identifies the variable type for all variables. Each element of any array is listed separately.

XRENUM—An enhanced program line renumbering facility which allows specification of an upper limit of the block of lines to be renumbered, supports relocation of renumbered blocks of code, and supports duplication of blocks of code.

XFIND—A cross reference facility for key words and character strings, also includes global replacement of keywords.

XCOMPRESS—Compress your BASIC programs to an absolute minimum. Removes extraneous information; merge lines; even deletes statements which could not be executed. Typically saves 30-40% space even for programs without our REM statements! Also results in 7-10% improvement in execution speed

ENTIRE PACKAGE FOR MODEL II $200
ENTIRE PACKAGE FOR MODEL III $125

AUTOMAP

Save time creating a formated screen with our Extended Basic Mapping Support system. The programmers task of communicating and displaying information with the user operator is completely automated with simple SEND and RECEIVE statement commands. Many different types of information can be sent and received from the operators with the use of a single command. Many hours of programming time are saved and cost efficiency is truly realized using Automap. $75

SUPERSNAPP X

You can sort fast using your present facilities, or you can do it faster with Racets' superb facilities, or you can use the fastest: SUPERSNAPP X. The heart of SUPERSNAPP X is a SUPER FAST in-memory sort routine that has been benchmarked against everything on the market and beats them all . . . hands down.

SUPERSNAPP X is the most-important component of SNAPP X EXTENDED BUILTIN FUNCTIONS which is a much needed set of additions to the Model II BASIC interpreter that will put time saving power at your fingertips. Let's compare (using random data) SUPERSNAPP X and Racet's GSF SORT for speed:

SORT	SUPERSNAPP X	RACET GSF
10,000 integers	39 seconds	59 seconds
5,000 Singles	22 seconds	34 seconds
2,000 Strings	10 seconds	15 seconds

SUPERSNAPP X is guaranteed to be the FASTEST in memory SORT on the market or your money back. With it you also get these EXTENDED BUILTIN FUNCTIONS: PEEK, PEEKW, POKE, POKEW, XDATS, XTIMS, ETIMS, FILES, AND THE SPECIAL SCMD (SNAPP___COMMAND). PLUS: open "E" Set SCROLL PROTECTION. ERASE all ARRAYS in one command. Specify size and Blink rate of CURSOR. LONG ERROR MESSAGES. Read from Video Screen Read. PEEK complete strings from memory. POKE complete strings to memory. convert upper case to lower case and vice-versa, turn complete screen off and on at will, extract largest or smallest values from user supplied list of numbers

We consider this next function as probably the most powerful addition ever made to Microsoft BASIC. **PRINT USING INTO A STRING.** The ability to arrange data into a string variable should perk your imagination.

Model II $100.00
Model III $75.00

SPOOLER Model I, Model II and Model III

Our workhorse! Unlike the one supplied with TRSDOS 2.0, ours requires no special knowledge or training on the part of the operator. Additionally, ours performs much better. On the Tandy SPOOLER, every time a disk is accessed, the printer stops dead! This package is available for Model I, in the TRSDOS/NEWDOS 80 versions, or for the Model II or Model III. Greatly enhances system performance when running typical business applications. Many applications have been benchmarked to run nearly TWICE AS FAST with the SPOOLER installed. Installs in minutes and no changes are required to your programs. The Model II version requires NO user memory. If purchasing more than one system, call for price quote. $100

SNAPP SCREEN

Another dependable product from Snapp, Inc. Using rigid plastic, Snapp, Inc. designed the ultimate green screen at the lowest price. The Snapp Screen installs in a Snapp with the use of pressure sensitive tape. Reduce user fatigue with the Snapp Screen. $10

DOUBLETAKE 3741

This is not a football play but the way to play ball fast in converting IBM 3741 and similar formated diskettes to Radio Shack formated disks or vice versa. Fast is the name of the game. $200

AUTOFILE

Another big time saver from Snapp, Inc. Autofile offers all programmers much needed time saving tools to use when working with direct files. Eliminate tedious commands such as LSET's, MKIS's, MKDS's, MKSS's, CVD's, CVI's and CVS's. Even the FIELD statement has been eliminated. You won't have to guess as to where the FIELDed variable is. The ASC and CHR$ function references will be performed automatically. $50

VSAM

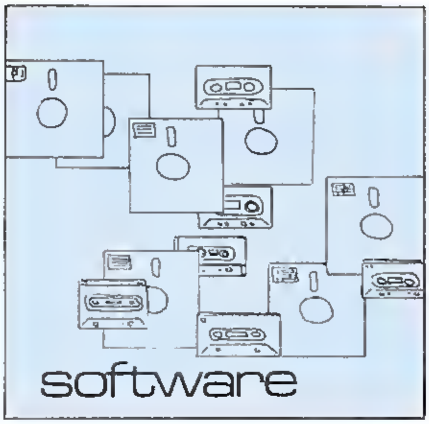
Virtual Snapp Access Method. V.S.A.M's development was built upon several man years of research and development by Snapp, Inc. studying all previously available access methods to small computers. What evolved was the ultimate keyed data access method available for any small computer. All records in the Data Base are truly variable in length to maximize the utilization of valuable disk space on TRS 80 systems. Irrelevant blanks are never physically recorded on the diskettes leaving only necessary data actually stored. Data Bases are defined with one primary key. Retrival with this key is quick and CONSISTANT without regard to maintenance activity levels. Additional fields may be described as alternate key fields. Access time using alternate key fields will be slower than through the primary field but it also will be CONSISTENT without regard to Data Base maintenance activity levels. VSAM is fully integrated into BASIC with commands easy to learn and use. $500.00

DOTWO

Allows programmer to build a list of any commands. All languages are supported. With DOTWO a programmer is now allowed to add any new commands of any kind and not be limited to subset restrictions. Anything that can be typed into the keyboard makes an acceptable command with DOTWO files. A programmer can virtually build an unlimited series of commands and not be limited to just TRSDOS commands. Included at no charge; BUILD/BAS which gives a programmer an easier method of creating and maintaining DOTWO or DO files. $100.00

HOSTII / TERMII

Allows remote control of a Model II from another Model II, or any ASCII terminal. Our Host system, unlike the one supplied with TRSDOS 2.0, supports accurate screen positioning on the Term station. Without this feature, formated displays appear on the screen looking like randomly placed garbage. Requires NO user memory! This system is designed to provide software support to our customer locations without ever leaving the office. Custom versions are now available for most nationally distributed terminals as a $25.00 option. Call for details. $50



Midway Campaign
Avalon Hill Game Co.
Baltimore, MD.
$15

by **David Tinis**

Midway Campaign is one of those games that on the surface seems laughably simple but proves, through playing, to be anything but. The creators of Midway Campaign, The Avalon Hill Game Co., are well-known for their excellent board games. Having already produced a board Midway game, a computer version seemed natural.

With all the graphics in use today, the lack of them in Midway Campaign is striking. Play is text oriented with the only graphics being a 12 × 12 grid of dots. These portray a map of the Pacific Ocean around the island of Midway. The two American task forces and Midway are under the player's control. The computer controls the three Japanese naval groups.

At the beginning of the game, the American units are placed in their historic positions; the date is June 3, 1942. Japanese forces are on the map but not shown. Their postion and composition is not known until they are spotted by search planes from Midway. Even though the American forces are visible to the player, the computer does not know where they are. It, too, must perform searches to locate the enemy.

At the beginning of the game, the player issues a Fleet command. There are four Fleet commands that display the map, the status of American aircraft carriers, change the heading of the task forces, and conduct aircraft operations. The computer remains in an interactive mode until an integer number is entered. This represents the length of time (in hours) the player wishes to play.

Next, the computer takes over. It moves American forces in accordance to Fleet commands, decides upon and executes Japanese actions, conducts searches and combat (if any) and checks for the end of game. Unfortunately, this can be rather lengthy and with no graphics involved the player has no recourse except to sit and wait.

Should the requested time pass or a significant event such as an attack or spotting occur, the computer returns to the interactive mode. If an attack must be resolved the computer does it prior to permitting the issuance of new Fleet commands.

As in the actual battle, the events are aircraft oriented. Search, we are told, is conducted by American PBY's from Midway and float planes from Japanese cruisers. However, we never see the search being conducted. We just sit and wait until told if there were any results. Likewise, when airstrikes are launched, we wait until the results are relayed. This waiting, in my opinion, is the game's major flaw.

Tactical Decisions

There is, however, enough in the way of realistic decision-making to keep the game interesting. Should the task forces be kept together or split up? How many fighters should be kept for defense of the carriers and how many sent out with the strike group? How many attack aircraft of each type in the strike group? Should all available aircraft be sent in hopes of getting in a crushing blow or should several waves be sent in? These tactical decisions will have a great bearing on the outcome of the game. A minor oversight can result in a sunk carrier.

The computer plays a very good game as the Japanese commander and is not easy to beat.

Midway Campaign is written in BASIC for the 16K Model I Level II TRS-80 and is available only on cassette. Along with the cassette are four pages of rules, historical background, examples of play and instructions for loading and running the program.

I would recommend Midway Campaign to those new to conflict simulation. Advanced game strategists will probably tire of it quickly. ■

ZBASIC Compiler
Simutek
Tucson, AZ
16K/32K Level II
Cassette and manual $79.95

by **Bruce Douglass**

Simutek has recently been advertising a new compiler called ZBASIC. ZBASIC comes in two packages; the first contains 16K and 32K versions for tape storage based systems and the second has 32K and 48K versions for disk based systems.

Pros and Cons

On the positive side is the small size of the compiler, the awesome speed with which it compiles, the run time speed of the programs it compiles, the interactive nature of the compiler, the fact that you can compile a 4K BASIC program in 16K, and that the compiler is *yours when you buy it* (no royalties to pay!).

ZBASIC sits in a fixed block of RAM, and its ORG depends on the version you use. The 32K version resides from 8680H to 9580H. Section 9200–9580 contains the subroutine package that gets tacked onto your program and speeds compiling as well as increasing run time speed. You could conceivably move the compiler anywhere you like by using an editor/assembler. I attempted to do this, using Radio Shack's EDTASM, and quickly ran out of text buffer. With a better, disk-based assembler, this should not be a problem.

Faster Than a . . .

The program compiles faster than a speeding daisywheel. Short programs compile instantaneously. I compiled a 2.5K BASIC program, and compile time was less than five seconds. I'm impressed with the speed of this program!

The ZBASIC manual lists run times for various commands in BASIC and ZBASIC. The increases in speed are from six times faster for A$ = INKEY$ to 12 times faster for SET and RESET and up to 288 times faster for jump commands such as GOTO and GOSUB. Using the SET command in a tight double loop, Level II BASIC requires 50 seconds to white out the screen. In its compiled form, it takes three seconds!

ZBASIC syntax is slightly different than Level II. A very nice feature of the compiler is its ability to jump back and forth between the BASIC program, ZBASIC, and DOS, enabling you to check for syntax problems as you tidy up your debugged BASIC program.

The compiler checks for errors during compile time and if it finds one, will return the type of error and the line number where it occurs. If you attempt to jump to a non-existent line, you get a peculiar error message, like line error in line 67757, which may cause some consternation if

NEVER UNDER$OLD.

Thats right. If you can find a lower price in this magazine WE WILL BEAT IT! Period.*

SPECIALS

SPECIAL #1 .... $44.50
TRS-80 Disk and Other Mysteries Book, Box of Verbatim Diskettes, and a Plastic Library Case.

SPECIAL #3 ...... $149
Apparat NEWDOS/80, Box of Verbatim Diskettes, and a Plastic Library Case.

SPECIAL #4 ...... $190
Microsoft Basic Compiler, Box of Verbatim Diskettes, and a Plastic Library Case.

SPECIAL #5 .... $49.95
Microsoft Basic Decoded and Other Mysteries, Box of Verbatim Diskettes, and a Plastic Library Case.

PRINTERS

EPSON MX-80 .... CALL
MX-80 Graphics ROM . CALL
NEW EPSON MX-80 TF CALL
Okidata Microline 80 .. $415
Okidata Microline 82 .. $625
Okidata Microline 83 .. $899
IOS Paper Tiger 460G $1050
Anadex DP-9000,1 .. $1195
Anadex DP-9500,1 .. $1295
Centronics 737 ..... $689
NEC Spinwriter ...... $CALL
NEC w/ Sellum Option . $CALL
Starwriter 25 cps ... $1440
Starwriter 45 cps ... $1770
Malibu 165 .... $1995
MPI 88G $495

TRS-80 HARDWARE

TRS-80's .... $CALL
Micropolis 77 track ... $399
Percom Doubler $199
Percom Separator $27
Orchestra 80 .... $79
Shugart 40 tr. $299
Pertec 40 tr. $299
MPI 40 tr. $299
16K Memory Kit ... $19.95
Cat Modem $145
D-Cat Modem $155
Lexicon Modem $125
Verbatim 525-01 .. $26.50
Verbatim 8" DD $44.00
Isolator ISO-2 ...... $49.95

NEW! Double Density Special

A Percom Doubler, Apparat's NEWDOS/80, Double Zap II [to convert the NEWDOS/80 to single and double density], a box of Verbatim Datalife diskettes, and a Plastic Library Case. A $436 Value...for

$359

Send For Our New Free Catalog!

*Include $2.00 shipping and handling on all orders. Visa and Master Card accepted. Never undersold offer applies only to items currently deliverable from other retail vendors at advertised price. All never undersold offers good as supply lasts. Please add $2.00 for all COD orders. Please call for items not listed. We gladly answer any questions on all of our hardware, software, and supply needs. Quanitity discounts available. School purchase orders accepted. Please remember to figure competitors shipping and handling charges when arriving at never undersold price.

POWERFUL SOFTWARE

Racet's New DISCAT $45
This New machine language disk catalog program will help you keep track of your programs in a neat, orderly library. It can catalog up to 1900 programs in each of 9 categories for a total of 17000 program location records.

Snapp Spooler $90
Fully automatic print spooler for the model I Increases your computers efficiency by freeing the computer while printing.

TRACKSESS $24
This is the complete disk zap utility, capable of copying standard and non standard formatting. The dynamic editing allows you to change any byte in any sector with amazing ease

SCRINPUT $29
Format your screen displays and input routines with this machine language subroutine. Gives your program a more professional look.

WORD PROCESSING

SPECIAL DELIVERY $118
The best in mailing list programs for the TRS-80 Models I & III. Special Delivery is easy to use & is written in fast machine language. Special Delivery allows you to merge your mailing list file with text files written with the Electric Pencil* or Scripsit* to create personalized form letters.

XTRA SPECIAL DELIVERY $189
Does everything special delivery does and more. Extra allows you to include special keyed messages in your text file, and includes Mailsort which allows sorting on multiple fields.

PROSOFT $79.95
"PROSOFT" Word processing software is written for the TRS-80 and the Centronics 737. Takes full advantage of all the capabilities available on the 737 including true proportional spacing. Comes complete with a 110 page users manual.

213-883-8594

ALPHA BYTE STORES

69

5115 Douglas Fir Road
Suite B
Calabasas, California
91302

the highest line number in your program is 200. This is the only case in which you get this kind of error message (as far as I know).

The manual lists some potential causes and corrections of errors. If you get an error message giving a line number that doesn't exist, try renumbering your program in BASICR or with a renumber program in high memory, and you will be able to locate the problem.

Another advantage of this program is that you can compile a 4K program in 16K. Note, however, that you can only compile for a 16K machine using the 16K version, as the compiled program resides in a fixed block of RAM and the larger versions won't run in 16K.

Mine and Mine Alone

Finally, I like the fact that when I buy this program, it is mine and mine alone. I don't have to pay anyone any money (except Uncle Sam) when I sell programs I write with ZBASIC, unlike Microsoft's compiler.

This is not the ideal compiler for all applications. It only handles integers, does not work with arrays, limits your variables considerably, does not support all Level II commands and starts and compiles into fixed RAM locations. Also, there is an error in the disk-saving version (although when I called Simutek they quickly advised that they would send me a new tape since mine was apparently defective).

Part of the reason the compiler works so fast is that it only handles integers. This is also why the compiled programs run so quickly. Multiple precision takes time and memory space; the compiler would have to be larger and slower to handle non-integers. The fact still remains that you need greater than integer accuracy for lots of applications. I was hoping to compile a 10K sophisticated multiple linear regression program that I wrote, but it cannot be done with ZBASIC. So assess your needs before you purchase this compiler.

The compiler also doesn't handle arrays. They can be simulated by fixing a block of RAM somewhere and using PEEK and POKE to store data. This requires some thinking on the user's part and rewriting most programs (all my programs use arrays!), but the method works. I compiled a program that required an array of 2000 elements, so I set aside 2000 bytes of memory. This is considerably more dense storage than using array variables anyway, and can often be used to save space in BASIC programs. Writing complex matrix calculations using this kind of data structure is not my idea of a good time, however.

Normally in Level II you have many variables: AA to ZZ, where the second letter can be anything from any letter to a single digit number. ZBASIC uses fixed RAM locations for its variables and limits your variables to save space. You may use 26 string variables, A$ to Z$, each of 31 characters. If the string is longer, you will overwrite the string above it. Thus, if LEN (A$) = 60, you will have wiped out B$. Numerical variables run from A-Z, A1-Z1, and A2-Z2. This is a fair number of variables, but it is inconvenient to rewrite programs changing all instances of several variables. In fact, it can be a real pain.

Some Level II commands are not supported, and the manual contains a long alphabetical list of these commands. They include VARPTR, SIN, COS, LOG, LEFT$, STRING$, CLEAR, and RESUME. The manual does give short routines to simulate SIN and COS (they return the value times 1000) and various string functions, including MID$, INST$, and RIGHT$. The meaning of some other BASIC commands are changed slightly as well. For example, you cannot use the logic operators *and* and *or* in conditional if. . . then statements, and if you use logical math operators, syntax must be closely watched.

One major fault of this program is that it fixes the RAM locations of its compiled programs. It would be so nice to be able to use the 32K version to write programs for a 16K machine. In the 32K version, the program begins at 9200H (the subroutine package); 846 bytes later, the compiled version of your program is tacked on. Variable memory for the 32K version begins at BC00H. To move the program to another memory location is a lot of work, using a disassembler and an editor/assembler.

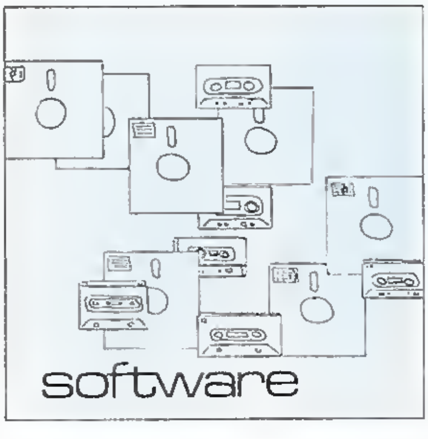
I spent 15 hours in an unsuccessful attempt to move a program (3.3K compiled) down to 4300H. It is tedious work and error-prone. If you have a very good disk-based assembler (not Radio Shack's), much of this misery can be avoided, but it is still not fun! A programmer at Simutek advised me that they do have a patch, but presently, it's unpublished. It will move the ZBASIC compiler down into low RAM, put the BASIC program up in high RAM, and will compile a 16K program that will run in a 16K machine, but the cost will be about $230 (and you must have ZBASIC also). He advised that it may sell for less once on the market; since it was new, the price had not really been decided upon.

On the version I received, the compiler made errors when it tried to save the programs onto disk. DOS would return with an Illegal Access Attempted To Protected File error. The file name would be put into the directory, but the program would not. After some work, I discovered that the second attempt, with the same file name, would save the program, but somehow the transfer address was messed up. The programmer at Simutek advised me that I must have a defective tape, and that they would replace it.

The manual is okay, but not up to the standard I would like in a program costing $130. It does give valuable information, but I don't feel it goes far enough. An alphabetical listing of commands supported by ZBASIC (they are listed in an apparently random fashion), an index, a more complete explanation of syntax differences and error codes, are all things I feel the manual should include. It does show how to link compiled programs (NEWDOS only), and gives some routines to get around or simulate normal BASIC functions in ZBASIC, and gives a memory map for the different versions, including the various fixed RAM locations for the variables. A complete map for the subroutine package would really be nice, so that you could easily link different machine language programs together with your ZBASIC program. As it is, you are stuck using USR. You must use the Level II USR format even with the disk versions. It is relatively simple to link your programs. However, with the subroutine map, you could write simplified programs to append to the compiled program, and call on the present subroutines.

I am impressed with several aspects of ZBASIC. Its most unfortunate aspect is the fixed memory ORGs for compiled programs. Lack of higher precision arithmetic can be a major problem. ■

Note: The price of the compiler was recently reduced from $130 to $89 for the 32K-48K disk version and from $99 to $79 for the 16K-32K tape version. The author considers this price to be more reasonable, and regards it as an extra plus in his recommendation of the product.



THE ALPHA I/O SYSTEM

A COMPLETE FAILURE?

It happened 3 years ago, when our President made a decision. At the time we specialized in custom analog and digital circuit design. The decision was to attempt to develop a line of standard interface hardware for the emerging microcomputers. At the time (1977) we had to decide which of the new machines could become the "industry standard" of the low cost micros.

Despite a few aggravating but minor deficiencies, the TRS-80 seemed tohave the most chance of success and it had the best price/performance ratio. Also, with some imagination, their large sales organization could become the largest service network in the world, a reassuring throught for the many novices in this new field.

It became clear that the TRS-80 could be used (with our then hypothetical system) to solve problems in many fields where computers were not yet used, mostly because of their high cost.

The IDEA was simple! ALPHA PRODUCT would supply the missing link between the TRS-80 and the "outside world", (more about this "outside world" later).

DANGER! If Radio-Shack entered the same market, we probably would not have survived, but the expectation was that they would be too busy developing their basic line (drives, printers, modem etc.). Thanks to our more specialized products, we would not be competing with them.

BAD START! We began with a failure. Our first product was supposed to be a simple, low cost, general purpose device. It would allow the TRS-80 to accept inputs other than the keyboard. Many kinds of external devices (the "outside world" mentioned before) like photocells, sensors, thermostats, switches, contacts, etc., could be connected easily. In addition, there were two relays to control (on or off) external loads such as motors, lamps, appliances, heaters, etc., etc.. In other words, it would allow the computer to interact or interface with external devices. We called it the INTERFACER 2. What a mistake! It sounded too much like "expansion interface". Many enthusiastic TRS-80 users called thinking that our "INTERFACER 2" was a low cost Expansion Interface (at $85 that would have been a real bargain!). We wanted to change the confusing name. That meant reprinting the manual, changing the ad, scrapping the flyers, discarding the silk screened cases. Well, "INTERFACER 2" it would stay.

TROUBLE! We also found that the majority of TRS-80 users were AFRAID of the hardware. They could be very comfortable with fancy programming but thought you had to be a computer specialist or technically inclined to put the INTERFACER 2 to work. In truth, some IMAGINATION and a SCREWDRIVER is all you really need. Anyone able to wire a switch could use this device.

WORSE! There was also the fear of plugging a "foreign device" into the precious computer. This notion has all but disappeared as there are now so many quality products designed for the TRS-80 that plugging in a non Radio-Shack device has become common.

Our ad in Creative Computing (80-Microcomputing did not yet exist) hardly paid for itself. We had a decision to make. Were we wrong or just too early? Our first INTERFACER 2 was sold to someone who wanted to, and succeeded in, controlling his fancy model railroad with his TRS-80. Interesting, but what made us stick with the concept was that some of our INTERFACERS began finding use in applications with fascinating possibilities. Space is lacking to describe them, but the most exciting was the successful use of the system in assisting a handicapped young boy. We were pleased to hear of such a meaningful application.

Three years later, as you can see in our ads, The INTERFACER 2 is alive and well. The price went up a bit, and despite the introduction of the more powerful INTERFACER 80, the sales have been steady.

Then came the least understood product! the ANALOG 80. This $139, nicely designed module is an Analog to Digital converter with 8 input channels. Used with your TRS-80, it provides a powerful "data acquisition system". This jargon simply means that you can monitor, measure and record 8 independant varying voltages. Very few people realized its real power. Such a system would have cost over ten thousand dollars just a few years ago.

The possibilities in scientific and engineering environments are endless. This system could replace chart recorders, digital data recorders, programmable calculators, data analyzers and many other specialized and expensive pieces of equipment. Furthermore, up to 8 ANALOG 80's could be used simultaneously for a total of 64 channels of analog input! They simply plug into the TRS-80 using our "X" series of bus extenders (EXPANDABUS).

Our next product was to be a second generation, Input/Output interface, with more flexibility than the INTERFACER 2. Careful design and refinement yielded the INTERFACER 80, the most powerful real world interface on the market today. It has 8 inputs, each optically-isolated and 8 outputs, each with a relay contact. The INTERFACER 80 is fully compatible with our ANALOG 80, allowing these to be used together in order to create systems that control external devices based on "sensed" input under control of the TRS-80.

A FAILURE! in spite of our extensive advertising, very few are aware of the existence of the powerful ALPHA I/O SYSTEM.

THE FACTS ARE:

- The ALPHA SYSTEM/TRS-80 combination forms an incredibly versatile and powerful tool for acquisition/processing/control.
- In spite of its moderate cost, the system is sophisticated and reliable.
- The entire system can be easily programmed in BASIC using INP(X) and OUT X,Y commands.
- The modular approach and our EXPANDABUS allow for instant expansion as requirements demand.

The following pages contain more information about the devices mentioned here. We invite you to call or write to discuss your particular application.

TIMEDATE 80

WHY LOSE PRECIOUS TIME ?

Neat, Compact Design
3 Years Battery Life

Slips Inside E/I
(Y Option Shown)

Real Time Without
Expansion Interface

•Complete, self contained "true" real time clock/calendar, TIMEDATE 80 continues to keep accurate time and date when the computer is turned off or experiences a power failure.

•TIMEDATE 80 only needs to be set once, and it's two replaceable "AAA" batteries (not included) keep TIMEDATE 80 running in excess of 3 years. Costly Ni-Cad batteries and charging circuits are eliminated.

•The instant power is applied to the TRS-80, TIMEDATE 80 provides MO/DATE/YR, DAY of WEEK, HR:MIN:SEC and AM/PM information with quartz accuracy.

•TIMEDATE 80 replaces the computer's internal clock. Extremely useful for automatic operation of remote systems with no operator in attendance. If the power fails and then is restored, only TIMEDATE 80 will update the system with current TIME and DATE information, an impossibility with the computer's internal clock.

•TIMEDATE 80 is quartz crystal based with INTELLIGENT CALENDAR, including provisions for leap year! TIME display may be by 12 hour AM/PM or by 24 hour military and Eruopean format.

•TIMEDATE 80 plugs directly into the rear of the TRS-80 keyboard and gives the "TIMES" function even without an Expansion Interface. For those with a disk system, it plugs into the left side panel of the Expansion Interface. An optional "Y" connector can provide for further expansion.

•TIMEDATE 80's small size keeps the computer table uncluttered. If you have an Expansion Interface, TIMEDATE 80 literally "DISAPPEARS" by slipping into the empty space in the bottom of the interface.

•Two sets of software, on cassette, come with TIMEDATE 80—"TIMESET" and "TIMES". "TIMESET" is a step by step set of simple instructions for setting TIMEDATE 80. "TIMES" is a set of poke routines which patch DOS and Level II TIMES to read TIMEDATE 80 and is easily incorporated into any user software. "TIMES" will always print the time and date when LISTING a program—great for keeping track of revisions!

•Other valuable uses for TIMEDATE 80 are: accurate date and time information for business reports like payroll records, financial reports, etc., or to various I/O devices requiring 24 hour clock input, such as laboratory instrumentation, and to communication systems needing "Log In/Log Out" data (bulletin boards).

•TIMEDATE 80, fully assembled and tested, 90 day warranty, complete with instructions and software on cassette, $95.00. "Y" option, add $12.00

PRINTER STAND

$59.95

PRINTER CABLE EXTENDER

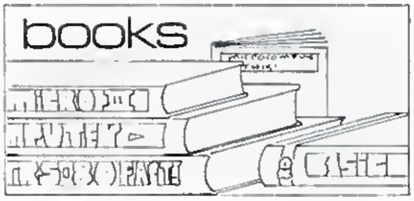
Adds 4 ft. to your existing cable, printer extension connects between Exp. Int. and your present printer cable........$27.50

ALPHA Product Co.

208

ADD $2.50 PER ORDER FOR SHIPPING AND HANDLING
ALL ORDERS SHIPPED FIRST CLASS MAIL
WE ACCEPT VISA, MASTER CHARGE, CHECKS, M.O.
COD: ADD $2.00 EXTRA.
QUANTITY DISCOUNTS AVAILABLE.
N.Y. RESIDENTS ADD SALES TAX.

85-71 79 St. WOODHAVEN N.Y. 11421 ☎ Info and order: (212) 296-5916



Owning Your Home Computer: The Complete Illustrated Guide
Robert L. Perry
Everest House
New York, NY
Softcover, 224 pp.
$10.95

by Mary Shooshan

Oh, no, another "complete" guide. It's truly amazing how the total knowledge on a subject can be reduced to 224 pages (counting 24 pages of appendix, glossary, bibliography, and index).

Have you ever noticed how these "complete" books always start with chapter(s) on history, miscellaneous stories and trivia? This one has an incredibly technical and involved chapter on microcomputer networks. (Specifically, computers connected by telephone to other computers.) It would have made a good appendix—at the end of the book—after you know something about computers, not before.

An Eggbeater for the Brain

Non-technicians, do not fret. Just skip over to chapter two, which is written just for you: "What is a Home Computer?" Did you know that the computer is a "mind appliance"?—an eggbeater for the brain. Don't worry that the computer will be smarter than you, since "any computer, no matter how large or small, is an idiot. It's dumb, stupid, inert until you tell it what to do." But, of course, you want one in your home.

Seriously, after this initial fooling around, Perry does settle down on page 27 to talk about the parts of a microcomputer and the common terms: input, data, CTR, etc. He also devotes many pages to the various systems on the market, emphasizing those with exciting sounds, colors, music, and graphics—things that whirr, beep and go bump in the night.

However, his information is vague and incomplete. He doesn't mention Radio Shack's Model III, which is replacing the Model I, or their Color Computer. He confuses hardware and software advantages, implying that some systems are user oriented (easy for non-programmers to use) when that often depends on the software.

One chapter is entitled, "99 Things to Do with a Home Computer". Unfortunately, it is not much more than a description of ninety-nine programs (out of the millions) on the market.

Perry talks about computer uses in education, especially home education, in aiding the handicapped, and in business. He gives some useful suggestions for finding a good microcomputer for your business and even gives a lesson in BASIC computer programming (remember, this is the *Complete Guide*), but even here there are errors in his information.

Perry states that the command, PRINT HELLO, will cause the computer to print Hello on the screen. However, this is not true. The computer will attempt to analyze Hello as a number (the number zero). If you want it to print a word, you must use quotes: PRINT "HELLO". If you want to learn programming, get a book written just for programming.

One of the main thrusts is the future applications of the computer. Specifically they are the computers that can tie in through modems and telephones to larger computers and networks of computers for communications and sharing information and programs; and computers that will run the house—doing everything from controlling the thermostat and locking the doors to watching the kids and feeding the dog. Tying in to a network will provide a lot of information at low cost once there are enough microcomputers around to make a network practical. This might be something to keep in mind for the future. As for computers running the house, microprocessors (the brain of the computer) are finding their way into many appliances, such as microwave ovens.

Attempts Too Much

To sum up, some of Perry's information is useful and helpful for laymen, some is interesting for technical people, and some is inaccurate and confusing. The book attempts too much; it tries to speak to all audiences on all aspects of microcomputer development and use, and it reaches none. It is poorly organized and difficult to read. As a "complete" book, it has something for everyone, but, unfortunately, it does not have much for anyone. ■

B-1 Nuclear Bomber
Nukawar
North Atlantic Convoy Raider
$15 each on cassette
Avalon Hill
Baltimore, MD

by Bob Liddil

B-1 Bomber, Nukewar and North Atlantic Convoy are among the better new games released in 1980 by Avalon Hill.

Avalon Hill has long prided itself on high quality boxed simulations for the sophisticated gamer. The new computer games are boxed in book size. Each contains precise documentation, a vacuformed tape holder and an Avalon Hill catalog. The tape contains programming for TRS-80, PET and APPLE; one dump for each computer.

B-1 Nuclear Bomber

B-1 Nuclear Bomber simulates a manned bomber strike into Russia. This is a navigation simulation with a nuclear climax. There are reciprocating MIGS to consider, as well as surface to air missiles, and many air combat variables. There is more than ample entertainment in B-1 Bomber, enough to keep you coming back for more. Win or lose, a game summary is delivered in the end, so you have a chance to compare scores and strategies from different game times.

Nukewar

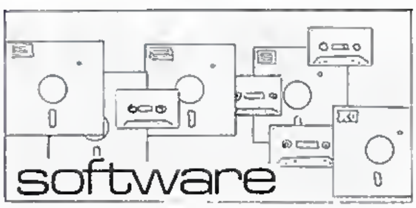
Nukewar is a bizarre simulation of a real life problem that faces world leaders every day: How to provide strategic defense for a country when faced with enough potentially destructive power to level the planet. During the Cold War, you are peacefully engaged in the building of new bases and, of course, spying on your neighbors.

Any country may declare nuclear war at any time. Negotiations for peace may ensue when the computer calls you on the hotline, wanting to make a deal.

North Atlantic Convoy Raider

North Atlantic Convoy Raider puts you at the helm of the mighty Bismark, pitting your battleship against the British. Fog, nightfall and pesky British warships all conspire to keep your Bismark from her destiny of destruction. It takes a while to win, but it's worth the effort.

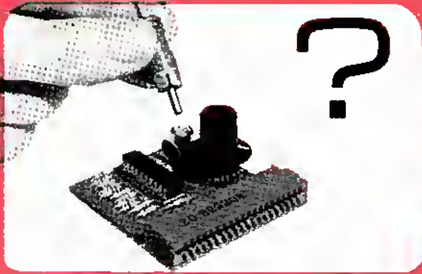
The best thing about the Avalon Hill games are their prices. In this day of $49.95 boxed computer games, Avalon Hill has priced their products nominally. ■



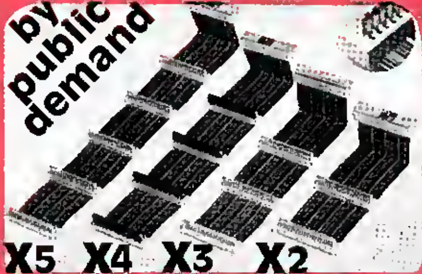
2printers?

Have two printers on line at all times and select printer 1 or 2 by software or built-in push-button. End the hassle of plugging and unplugging printer cables with our new PRINSELECT 80. $25.00

STICK-80 MAKES KEYBOARD OBSOLETE.
Features the famous ATARI Joystick 8 directions + fire control. Simple instructions to make joystick versions of most action games. Plugs into keyboard or expansion int... Price includes ATARI joystick with ALPHA interface and instructions. FREE "MAGIC ARTIST" program $39.95
Super Real Time Action Graphic Sound games for Stick-80 by Software Innovations ALIEN INVASION
COSMIC INTRUDERS BREAKOUT Each $9.95
STELLAR ADVENTURE super action with sound $14.95
Software authors and distributors contact us for joystick conversion package for your existing games



MUSIC-80 MUSIC-80 MUSIC-80 MUSIC-80 MUSIC-80
Use existing software or write your own. With this low cost 8 bit digital to analog converter you can synthesize up to 5 music voices. Built-in volume control handy when stereo not near TRS-80. Simply plug the "MUSIC-80" into the keyboard or the E/I screen printer port and connect the output (RCA jack) to any amplifier. The Radio-Shack $12 speaker/amplifier works fine.
Fully assembled and tested, 90 day warranty $39.95



YOU ASKED FOR IT: "EXPANDABUS" X1, X2, X3 AND X4.
CONNECT ALL YOUR TRS-80 DEVICES SIMULTANEOUSLY on the 40 pin TRS-80 bus. Any device that normally plugs into the keyboard edge connector will also plug into the "EXPANDABUS". The "X4" is shown with protective covers (included). The TRS-80 keyboard contains the bus drivers (74LS367) for up to 20 devices, more than you will ever need. Using the E/I, it plugs either between KB and E/I or in the Screen Printer port. Professional quality, gold plated contacts. Computer grade 40 conductor ribbon cable.
X2 $29 X3...$44 X4 $59 X5 $74
Custom configurations are also available, call us.

ANALOG-80: A WORLD OF NEW APPLICATIONS POSSIBLE.
8 DIGITAL MULTIMETERS PLUGGED INTO YOUR TRS-80!!!
Measure Temperature, Voltage, Current, Light, Pressure etc. Very easy to use: for example let's read input channel #4. 10 OUT 0,4 'Selects input #4 and also starts the conversion 20 A = INP(0) 'Puts the result in variable "A" Voila! Specifications: Input range 0-5V to 0-500V. Each channel can be set to a different scale.
Resolution 20mV (on 5V range). Accuracy 8 bits ± 5%). Port Address jumper selectable. Plugs into keyboard bus or E/I (screen printer port). Assembled and tested. 90 day warranty. Complete with power supply, connector, manual .$139

INTERFACER 2: LOW COST INPUT/OUTPUT MODULE.
Still the best value in sense/control devices. Use it for energy control, burglar alarm, darkroom, selectric drive, model trains, robots, Skinner box...
—8 latched TTL outputs, 2 relays SPDT 2A 125V, contacts
—8 TTL/CMOS inputs. Input 0 and 1 are optically isolated.
—Neat and compact design, very easy to use:
10 A = INP(0) 'Reads the 8 inputs (if A = 0: all inputs are low) 20 OUT 0,X 'Controls the outputs and the relays
Assembled & tested. 90 day warranty. Price includes power supply, cable to KB or E/I, superb user's manual, free phone dialer program: $95. Manual only: $5.

GREEN SCREEN WARNING

IBM and all the "biggies" are using green screen monitors. Its advantages are now widely advertised. We feel that every TRS-80 user should enjoy the benefits it provides. But **WARNING:** all Green Screens are not created equal. Here is what we found:

- Several are just a flat piece of standard colored Lucite. The green tint was not made for this purpose and is judged by many to be too dark. Increasing the brightness control will result in a fuzzy display.
- Some are simply a piece of thin plastic film taped onto a cardboard frame. The color is satisfactory but the wobbly film gives it a poor appearance.
- One "optical filter" is in fact plain acrylic sheeting.
- False claim: A few pretend to "reduce glare". In fact, their flat and shiny surfaces (both film and Lucite type) ADD their own reflections to the screen.
- A few laughs: One ad claims to "reduce screen contrast". Sorry gentleman but it's just the opposite. One of the Green Screen's major benefits is to increase the contrast between the text and the background.
- Drawbacks: Most are using adhesive strips to fasten their screen to the monitor. This method makes it awkward to remove for necessary periodical cleaning. All (except ours) are flat. Light pens will not work reliably because of the big gap between the screen and the tube.

Many companies have been manufacturing video filters for years. We are not the first (some think they are), but we have done our homework and we think we manufacture the best Green Screen. Here is why:

- It fits right onto the picture tube like a skin because it is the only CURVED screen MOLDED exactly to the picture tube curvature. It is Cut precisely to cover the exposed area of the picture tube. The fit is such that the static electricity is sufficient to keep it in place! We also include some invisible reusable tape for a more secure fastening.
- The filter material that we use is just right, not too dark nor too light. The result is a really eye pleasing display.

We are so sure that you will never take your Green screen off that we offer an unconditional money-back guaranty. Try our Green Screen for 14 days. If for any reason you are not delighted with it, return it for a prompt refund.
A last word. We think that companies, like ours, who are selling mainly by mail should •list their street address•have a phone number (for questions and orders)•accept CODs, not every one likes to send checks to a PO box•offer the convenience of charging their purchase to major credit cards.
How come we are the only green screen people doing it?
Order your ALPHA GREEN SCREEN today ..$12.50

LET THE "CHAIN BREAKER" FREE YOUR MINI-DRIVES.
End the daisy-chain mess once and for all. Fits all mini-drives: Percom, Aerocomp, Shugart, Micropolis, MTI, Vista, Pertec, Siemens, BASF. Easy to install: just remove the drive cover, plug in the "CHAIN BREAKER" and replace the cover. Voila!!!
Now you can change and move your drives around without disassembly. Keep the cover on and keep the dust out. High reliability gold plated contacts, computer grade 34 conductor cable. Tested and guaranteed.
Get one for each drive... only $13.95

8 power relays under your control

INTERFACER-80: the most powerful Sense/Control module.
- 8 industrial grade relays, single pole double throw isolated contacts; 2 Amp @ 125 Volts. TTL latched outputs are also accessible to drive external solid state relays.
- 8 convenient LEDs constantly display the relay states. Simple "OUT" commands (in basic) control the 8 relays.
- 8 optically-isolated inputs for easy direct interfacing to external switches, photocells, keypads, sensors, etc. Simple "INP" commands read the status of the 8 inputs. Selectable port address. Clean, compact enclosed design. Assembled, tested, 90 days warranty. Price includes power supply, cable, connector, superb user's manual. $159

✓210

ALPHA Product Co.

ADD $2.50 PER ORDER FOR SHIPPING AND HANDLING
ALL ORDERS SHIPPED FIRST CLASS MAIL
WE ACCEPT VISA, MASTER CHARGE, CHECKS, M.O.
COD: ADD $2.00 EXTRA
QUANTITY DISCOUNTS AVAILABLE
N.Y. RESIDENTS ADD SALES TAX

85-71 79 St. WOODHAVEN N.Y. 11421 ☎ Info and order: (212) 296-5916

Basic FORTRAN
James S. Coan
Hayden Book Company, Inc.
Rochelle Park, NJ
Softcover, 248 pp.
$8.95

by Dave Smith

While it may seem strange to see a text on FORTRAN reviewed in a magazine catering to users of BASIC, James Coan has packed a wealth of information into his text which can be of value to programmers in any of the high level languages. The text is designed for an elementary FORTRAN course, but lends itself equally well to the individual who wishes to learn FORTRAN on his own and as a supplemental text for a precalculus mathematics course. All you'll need is a knowledge of first-year algebra. Coan develops all additional algorithms required in the text.

There are two premises upon which *Basic FORTRAN* has been written. The first is that the reader is essentially a newcomer to computer programming; the second, that for a rewarding learning experience, the student should begin meaningful programming immediately.

Responding to premise one, the author clearly defines (on page one) his method for familiarizing the student with FORTRAN while simultaneously inculcating good programming habits. Not without humor, he draws the analogy between learning to program and learning to drive a car. He notes that, although each process can be accomplished from a book, both are facilitated by the availability of a machine on which to exercise the knowledge.

Having explained the method, Coan proceeds to implement it. The author presents the student with his first program on page two. Logically enough, this three-liner is a program to generate output from the computer printer, thereby giving immediate reinforcement to the novice programmer. Each line of the program is analyzed, and Coan gives the student only enough information to foster understanding without inundating him with related but non-essential detail. By increments of one and two lines, this initial program is expanded throughout chapter one until the three-liner has been developed into an 18-line program to compute and write paycheck amounts.

Chapter one closes with problems and a summary. Each type of statement introduced in the chapter is reviewed, as is each structural or programmatic concept. The problems exercise the material presented in the chapter, and an appendix provides answers and sample solutions to every other problem.

Language and Applications

While the chapter one format is basically that of the entire book, the author departs from this format by introducing problems and exercises more frequently chapter by chapter. He is careful to divide the material into readily digested elements and encourages the reader to become fully familiar with each new concept before continuing. The self-instructed programmer can easily establish his/her own pace for absorbing the chapters of the book and the features of FORTRAN.

Of the eleven chapters in the book, the first five are devoted primarily to the language of FORTRAN, while the remainder emphasize applications. There are more than 80 programs developed throughout the text. Applications range from simple business and finance to more complex data processing, graphing, quadratics, trigonometrics, polynomials, probability and random simulations.

One of the many attractive features of this book is that all sample programs are listed in an index, which facilitates review and permits the book to serve as a handy reference. Other appendices provide a table of FORTRAN-supplied functions and a Z-80 random number function (since most FORTRANs do not incorporate random number generation.).

As with *Basic BASIC* and *Advanced BASIC*, Mr. Coan has filled a small number of pages with a sizeable quantity of information, and has presented it so as to capture the interest of the broadest spectrum of students of FORTRAN. ■

Getting Started with TRS-80 BASIC
Going Ahead with Extended Color BASIC
Tandy/Radio Shack
Ft. Worth, TX
Softcover
$6.95 each

by Lynda Stratton

Hot off the presses from Radio Shack are two books designed for the new computer enthusiast.

Because I am a beginner myself, initially I was leary. I had visions of complex figures and hard-to-understand diagrams. However, being familiar with Radio Shack's flawless *User's Manual for Level I*, I thought perhaps I might be in for another treat. I wasn't disappointed.

Getting Started with TRS-80 BASIC and *Going Ahead with Extended Color BASIC* are not only complete and easy to understand introductions to your microcomputer, but are also the kind of books that make learning fun.

Some of you are so dedicated to your computer that it might be called an addiction. Well, it would be very easy to become addicted with help from either of these books. Most importantly, they are tools to help you use your computer effectively (instead of floundering around in the dark the way most beginners do the first few months).

If your computer isn't ready to go, you'll want to get it ready first. As it says at the beginning of *Getting Started with TRS-80 BASIC*, both are do-it-now books.

Going Ahead with Extended Color BASIC

This book is for those of you who have some knowledge of TRS-80 BASIC, and have itchy fingers to try out new and colorful things. *Going Ahead with Extended Color* starts off more or less where *Getting Started* left off; reminding you of important concepts and procedures along the way.

Going Ahead is split into three sections, making it easy to skip those parts you are already familiar with. If you're in an artsy mood, start with the first section. If you're ready for more complicated problems and want to play around with a few figures, the middle section is for you. The back section is for everyone—it has answers to the exercises, which appear throughout the book, along with worksheets and useful tables.

The central exercise, which runs through the book, is a house building project. I must admit, I got hooked on this program and couldn't wait until it was finished and ready to move into. Friends came by and were horrified to see the color of the smoke from the chimney. "What the hell are you burning, you've got pink smoke." They all wanted to pitch in and add their ideas. "Draw a garage! Open the door! Put drapes on the windows! One friend even wanted me to have an airplane circling overhead!

As this is a first printing, there are a few things that need to be ironed out. For example, on page 42, they tell you to change the PMODE in line 20 to 4. This isn't too serious as it is very obvious that the PMODE is set in line 5, so go ahead and change line 5. Another error is on page 156. In the answers to exercises four and five, line 135 tells the program to GOTO line 145. There is no line 145, so I deleted line 135, and it didn't seem to interfere with the rest of the program.

Another plus to this book are the note

GAME BREAKTHROUGH:

THE STICK 80

- ☑ FEATURES THE FAMOUS **ATARI JOYSTICK**.
- ☑ WORKS WITH ANY LEVEL 2 AND DISK SYSTEMS.
- ☑ PLUGS DIRECTLY INTO KB. OR E/I (EXPANSION PORT).
- ☑ INCLUDES SIMPLE, DETAILED INSTRUCTIONS TO MAKE JOYSTICK VERSIONS OF MOST ACTION GAMES.
- ☑ COMPATIBLE WITH **ANY** OTHER TRS-80 ACCESSORIES.
- ☑ FULLY ASSEMBLED AND TESTED. READY TO PLUG IN AND ENJOY.
- ☑ FUN TO EXPERIMENT WITH IN **BASIC**: SIMPLY USE INP(Ø).
- ☑ 90 DAY PART AND LABOR WARRANTY.

UNCONDITIONAL **MONEY BACK GUARANTEE**.Try your STICK-80 for 14 days. If for any reason you are not delighted with it, return it for a prompt and courteous refund (including shipping and handling). PRICE INCLUDES ATARI JOYSTICK + ALPHA INTERFACE + DETAILED INSTRUCTIONS + DEMO PROGRAM. ONLY $39.95

BIG FIVE SOFTWARE SUPER FAST MACHINE LANGUAGE ACTION GAMES

If you and your TRS-80 have longed for a fast-paced arcade-type game that is truly a challenge, then **SUPER NOVA** is what you've been waiting for. In this two player machine-language game, large asteroids float ominously around the screen. Suddenly your ship appears and you must destroy the asteroids before they destroy you! (But watch out because big asteroids break apart into little ones.) The controls that your ship will respond to are thrust, rotate, hyperspace, and fire. All right! You've done it! You've cleared away all the asteroids! But what is that saucer with the laser doing? Quick! You must destroy him fast because that guy's accurate! ♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦ **SUPER NOVA**®

The sound of the klaxon is calling you! Cruel and crafty invaders have been spotted in battle formation warping toward Earth at an incredible speed. Suddenly, your ship materializes just below the huge flock of invaders. Quickly and skillfully you shift right and left as you carefully fire your lasers at them. But watch out! A few are breaking out of the convoy and flying straight at you! As the whine of their engines gets louder, you place your finger on the fire button knowing all too well that this shot must connect—or your mission will be permanently over! With sound effects! ♦♦♦♦♦♦♦♦♦♦♦♦ **GALAXY INVASION**®

Your TRS-80 screen has been transformed into a maze-like playfield for this game. As your ship appears on the bottom of the screen, eight alien ramships appear on the top. All of them are traveling at flank speed directly at you! Quickly and boldly you move toward them and fire missiles to destroy them. But the more aliens you destroy, the faster the remaining ones become. If you get too good you must endure the wrath of the keeper of the mazefield: the menacing "Flagship". You must destroy him fast because, as you will find out, that guy's accurate! With sound effects! ♦♦♦♦♦♦♦♦♦♦♦♦♦♦ **ATTACK FORCE**®

With thousands of stars whizzing by you, your **SPACE DESTROYER** ship comes out of hyperspace directly under a convoy of aliens. Almost effortlessly, you skillfully destroy every last one. But before you can congratulate yourself, another set appears. These seem to be slightly more intelligent than the first set. Quickly you eliminate all of them, too. But your fuel supply is rapidly diminishing. You must still destroy two more sets before you can dock with your space station. All right! The space station is now on your scanners! Oh no! Intruders have overtaken the station! You must skillfully fire your neutron lasers to eliminate the intruders from the station before your engines run out of fuel and explode! With sound! ♦♦♦♦♦♦♦♦♦♦♦♦♦ **COSMIC FIGHTER**®

NEW

The second **Big Bang** has occurred and the galaxy is full of stray asteroids and meteors. As you look through your space port you see a belt of asteroids drifting across the screen blocking your path to the safety of the space station above. But be careful because meteor showers, exploding suns and invading aliens may strike your ship and send it hurtling back to ground level. How many times can you and your opponent maneuver through those obstacles before time runs out? With sound effects! ♦♦♦♦♦ **METEOR MISSION II**®

SPECIAL

INTRODUCTORY OFFER:

Order the STICK-80 with one or more games and choose one:

•FREE ALPHA GREEN SCREEN
OR
•FREE MAGIC ARTIST PROGRAM

IN ADDITION YOU MAY:
ORDER 2 GAMES AND TAKE 10%
ORDER 3 GAMES AND TAKE 15% OFF the games

All games are written in machine language and supplied on cassette.

EACH GAME IS AVAILABLE IN 2 VERSIONS

A (SIDE 1: LEVEL II 16K FOR MODEL 1
SIDE 2: LEVEL II 16K FOR MODEL 3 $15.95

B (SIDE 1: DISK 32K FOR MODEL 1
SIDE 2: LEVEL II 16K FOR MODEL 1
THE DISK VERSION SAVES THE HIGH SCORES ON DISK $17.95

• ALL THESE NEW JOYSTICK GAMES WILL ALSO WORK USING THE KEYBOARD WITHOUT ANY MODIFICATIONS.

GOOD NEWS: If you already have a non-joystick version of these BIG FIVE SOFTWARE games, send the original tape(s) with your STICK-80 order + $3 per game. We will send you a new Joystick version of your game(s).

ALSO AVAILABLE **DOUBLE-STICK 80:** 2 ATARI JOYSTICKS + ALPHA DUAL INTERFACE + INSTRUCTIONS + DEMO PROGRAM: $59.95

ALPHA Product Co.

✓209

ADD $2.50 PER ORDER FOR SHIPPING AND HANDLING
ALL ORDERS SHIPPED FIRST CLASS MAIL
WE ACCEPT VISA, MASTER CHARGE CHECKS M O
COD ADD $2 00 EXTRA
QUANTITY DISCOUNTS AVAILABLE
N Y RESIDENTS ADD SALES TAX

85-71 79 St. WOODHAVEN N.Y. 11421 phone: (212) 296 5916

Introduction to TRS-80 Level II BASIC and Computer Programming

Micheel P. Zabinski
Prentice-Hall
Englewood, NJ
Softcover, 182 pp.
$10.95

by G. Micheel Vose

Computer operation and programming is an intimidating subject to someone with no experience with computers. Those of us who have been around computers for awhile think of them as sophisticated tools, nothing more. We are no longer afraid of them. But the novice programmer is often simultaneously awed and frightened by the thought of programming the mighty computer.

The teaching of computer programming, therefore, becomes tricky. This is because there is a fine line between catching a student's attention and overwhelming him with jargon and detail. Some authors of computer instruction books have tried to overcome this problem by taking a lighthearted, humorous approach to their subject matter. Others have presented the material factually in a straightforward manner, letting the student learn if he/she can.

Michael Zabinski adopts yet another approach, although not a novel one. He takes a scholarly, textbook approach in his new book. This book would be suitable for use in a junior high or high school course in computer programming and would also be appropriate for college students or adult education students.

What makes Zabinski's book so acceptable is its liberal use of examples and a set of extremely challenging problems to solve using a computer. Even experienced programmers enjoy the pure challenge of solving good problems and this book has dozens. It has challenges for the quick learner but also is accessible by students at all levels.

Neturelly end Eesily

This book, while only 150 pages or so not counting the appendices, covers all the Level II BASIC commands and functions. Starting with the elementary stuff like turning on the system and getting it to print 'Hello', the text moves naturally and easily into a well written introduction on how the TRS-80 uses numbers. Most people I know are afraid of computers because they think their math skills are weak (that's why we invented the darn things!) and they imagine that computer programming requires top-notch algebraic ability. Zabinski has sensed this and introduces the subject of using numbers early but gently. Once introduced, the subject of math no longer seems so important and the text moves on to define numeric and string variables and the TRS-80 arithmetic functions.

At this point, the student has learned very little actual programming. So the book takes advantage of the student's still uncluttered memory and discusses program logic and the use of line numbers. From there the discussion proceeds through input statements, program line editing, debugging, flowcharts, IF...THEN, FOR...NEXT, READ... DATA, subscripted variables and all the rest of the TRS-80 commands and functions.

The Key

Of course, the only way to learn most anything is to do it. This author knows the student will be sitting at his/her computer while absorbing its lessons and makes liberal use of examples. The examples are set off from the text and are annotated, line by line. They are well chosen and illustrative without being too simple.

This is a well organized book, put together by a knowledgeable programmer/educator. The writing style is a little stiff, typical of a textbook, and some graphics would have brightened up its appearance. (There is not even a picture or a line drawing of a TRS-80, except on the cover.) The publisher chose to use the standard gimmick of a dot-matrix printer to simulate TRS-80 screen displays on the printed page (which I personally find annoying). However, the text is complete and informative. ■

pages for computations. It solves the problem of misplacing that little slip of paper with all those important numbers which took so long to figure.

Getting Started with TRS-80 BASIC

Getting Started with TRS-80 BASIC is for use with the Model I and II. It comes complete with do-it-yourself exercises, sample programs, tables, and worksheets. The book utilizes cartoons that make learning with this book fun. In addition, this book has what I call a bird with a smart-aleck grin on its face which bugs you and tells you about error messages and how to deal with them. This book provides a good foundation along with giving confidence to all beginners to get started on bigger and better projects.

The pages are uncluttered and spacious, the text has a leisurely attitude that impresses on the reader that computers are for everyone, and can help you do anything you want with the computer (well, almost anything). One warning: you may get so involved with this book you won't notice the passing of time. So tell your friends and neighbors not to be too concerned if they don't see you around for a few days.

At the end of each chapter are checkpoints, in the form of a short quiz, to make sure you've been paying attention. Don't skip these, they serve as useful reminders of methods taught in the preceding chapter. If you've read the chapter thoroughly, you'll find them easy.

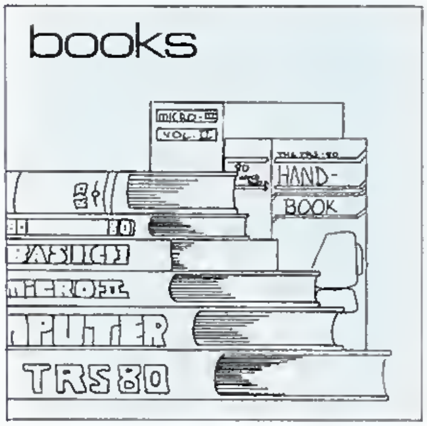
Getting Started with TRS-80 BASIC starts with the basics from setting up your TRS-80 and turning it on, to writing your own simple programs and debugging them.

It is split into three sections: Part one: "Getting Started," is for first timers who need to start at the very beginning: getting to know the keyboard, words and definitions, and printing messages. Part Two: "Basic Training," familiarizes the reader with BASIC, editing, strings, statements, shortcuts, math functions and arrays. Part Three: "Exploring the Territory," is the most advanced section and teaches graphics, error-handling and how to neaten printed statements (messages).

Both of these books are very enjoyable, and made learning easy. If you're not a computer enthusiast to begin with, there is the possibility that one or both of those books will turn you into one.

Going Ahead with Extended Color BASIC was my particular favorite, possibly because it had the added advantage of color (and at the ripe old age of twenty-five, I still can't resist a coloring book).

Whichever book you choose, I guarantee they will prove to be valuable companions to your keyboard. ■



From page 8

you if you bring him your problems and take your money elsewhere. The bottom line is, if you have a local authorized outlet from whom you can buy, do it!

There are some non-authorized folks selling TRS-80s. Some even advertise that you can take your purchase to any Radio Shack for repair. The non-authorized "dealer" was the original purchaser, and his warranty does not "pass through" to the person he sells to. Often these folks install their own RAM or disk drives or who knows what. My best suggestion to you is that you be prepared to return it to him for repair. We're not equipped to service non-Radio Shack drives; so be prepared for a possible service problem!

Line Printer VII Owners

We've found that the paper involved in mailing labels won't pull through the Line Printer VII's mechanism reliably. Any of you who were considering handling a mailing list on the VII (doubtful at 30 CPS), had better think about another printer.

Printing Multi-part Forms

Not too many of you will want to do this, either, but we did find one person, and we think he really ended up with a problem.

We've all seen (or used) multi-part forms which are bound together at the top with carbon sheets. Some of them get pretty thick. If they're too thick, they'll push the ribbon out of place on some dot matrix printers. If the ribbon moves, and the paper is thick enough, it could hook the print wires as they strike the paper. If the wires are bent or broken, you've bought an expensive new print head! Be careful what you try to put through your line printer.

Color Computer Memory

A couple of months ago, I wrote that the Extended BASIC Color Computer had about 14.5K of user memory. Well, in point of fact, the number is 13,095 bytes, plus 200 bytes reserved for string space. So, was I wrong telling you 14.5K? Well... almost. It seems that the guy from whom I got the number for my column was one of those creative programmers; and it seems there is another reserved area of memory for graphics, of 1,500 bytes. And it seems further that if you're creative enough (and desperate enough), you could store variable values there and retrieve them with PEEKs and POKEs, saving user memory.

If you're really going to die without another few bytes, read the thorough discussion of Color Computer memory in the June issue of our owner's newsletter, on the CC Product Line Manager's page. ■

80 ACCOUNTANT

by Michael Tannenbaum C.P.A.

Last week I received a call from one of my clients. In the process of completing a month's work on one of their new subsidiaries they were attempting to print an income statement when the program stopped with an FC error. Since the ledger program (Radio Shack's) had been running without a hitch in their other companies, I requested a backup so I could duplicate the error on my computer.

After reviewing the listing and the supplied data files, I found a coding error in the ledger account classifications. As a result, a counter was not initialized in the report generating program. The illegal function call was the result of an invalid FOR...NEXT loop which depended on the counter.

Who was at fault, the client who did not properly code the account classification, or the system designer who did not anticipate the problem? To the client the answer was simple; it was the computer's fault. He was right, the computer should have displayed a more informative error message. An FC error in line 4040 is meaningless. Yet how could a system designer anticipate every possible combination of input error?

The point of this lesson is that it's impossible to account for all errors. For this reason, Radio Shack and all other reputable software suppliers provide bug fixes for their customers. These fixes are published monthly in a newsletter which is sent to computer owners free for the first year after purchase.

Catching the Bugs

In addition, Radio Shack has offered to supply free bug fixes to registered owners of their software. (To register simply fill out the card in back of the software package.) It can save you lots of time and effort when that unanticipated bug hops out of the grey box.

Until recently, knowledgeable customers have been able to debug their own software. Each Radio Shack program has included a printed listing. Not only were these listings useful for tracking down software problems, but they were also full of good programming ideas.

Radio Shack's latest accounting software, however, is being supplied as compiled COBOL programs. While this permits Radio Shack to retain control over the code, it makes it very difficult to correct bugs or interface them with other programs. Fortunately, I have been told that source listings will be made available for a license fee.

Radio Shack is trying to catch as many bugs as possible before this software is released. New software is being tested by a user who is given extensive assistance by the vendor.

This method of pre-release testing is an accepted practice for larger software systems. Despite this, most commercial software firms recommend software maintenance contracts which cost approximately 10 percent of the system price.

Advantages of COBOL

The release of COBOL programs for the Model II is a surprising and interesting development. The advantage of COBOL (COmmon Business Oriented Language) lies in its wide application for most major computer systems. With the source listing, it is possible that a Model II accounting program could be compiled and run on a Hewlett Packard Minicomputer (the HP 3000 or the IBM 4331). The obvious result is to increase the market for the software which should mean better programs.

Conversely, some COBOL programs that are already in commercial use might be compiled on the Model II. This could be quite interesting; the amount of COBOL software available is enormous. This language has been used for over 20 years and COBOL programs are transportable between computers.

Because the software needs of science and industry are different, FORTRAN (FORmula TRANSlator) was developed for scientists and COBOL was developed for business users. BASIC, which is familiar to many of us, is a subset of FORTRAN and was originally designed as a FORTRAN teaching aid.

Both FORTRAN and COBOL differ from BASIC in a fundamental way. They require compilation before execution. The compiler is the program that is customized for each computer. Thus, theoretically, a COBOL or FORTRAN program can be executed on any computer that has a COBOL or FORTRAN compiler.

Yet, in practice this is rarely possible. There are enough differences between

computers so that some adjustment is usually required. In fact, in the very first demonstration of a COBOL program 20 years ago, code written for one computer, an RCA 501, required a bit of adjustment before it would compile on the other, a UNIVAC. The estimate, at that time, was that 10 percent of the code required adjustment. (Little has changed. This is still a good estimate.)

Another problem which could bar the use of commercial software is that the COBOL compiler used on the Model II does not support the ANSI 74 (American National Standards Institute 1974 version) COBOL. Although all exceptions are documented in the manual (catalog # 26-4703), these exceptions could make it quite difficult to run a system on the Model II which has been developed for a larger computer.

Why then has Radio Shack decided to release its new software as compiled COBOL programs? Our test of the COBOL Receivable program gave us a clue. This

"Price extensions, sales taxes and invoice totals are automatically calculated."

system executed far more quickly than the BASIC version I reviewed. In addition, the COBOL file structure apparently allowed Radio Shack programmers to provide for up to 1800 accounts and 4100 transactions on a three drive system. This should enable many more firms to utilize the Model II for receivable processing.

Radio Shack's COBOL receivable capacity is defined during the set up procedure. The file structure is expandable to accommodate growth. A single drive system can handle up to 400 accounts and 850 transactions. A two drive system will handle 800 and 4100 transactions, and a three drive system will handle up to 1800 accounts and 4100 transactions.

Once you've finalized the hardware configuration, running the set up program initializes the system. This program defines the anticipated file capacity, definitions, company name to be printed on reports and statements and general ledger account codes.

The system definition section devotes considerable attention to a schedule of finance charges. The package offers a stepped schedule, a minimum charge or a flat charge. These should be valuable to retailers who maintain their own open accounts.

Because the options selected at set up time automatically affect finance charge calculation, check your local statutes *before* using the system.

Accounts Receivable

The accounts receivable system integrates with the BASIC general ledger system. At the end of the month a program that extracts the accounts receivable totals creates a file that is accepted by the general ledger. When this file is created, the general ledger master file must be present. The GL master file validates the codes assigned to the receivables. For this reason integration is only feasible in a two drive system.

If an invalid account is encountered during the account transfer operation, the transfer is aborted and an error indicated. This ensures that only properly coded data is accepted in the general ledger. In an integrated system, therefore, your accounts receivable codes must be identical with those *already* established in the general ledger.

The accounts receivable system uses alphabetic codes to indicate accounts to be charged or credited. Then accounts are predefined and used as default codes. The first eight default codes—A through H—are permanently defined to ensure that processing will not be interrupted at an inconvenient time. The eight permanent codes are: accounts receivable; sales; cash in bank; sales tax; freight; finance charge; discounts; and non-taxable sales.

The coding features available in the accounts receivable system are quite flexible. With 26 codes available, a complex sales analysis can be accommodated. With the proper coding you might be able to eliminate the need for separate sales and sales return journals. Because of the potential time saved you should discuss this coding with an accountant at setup time.

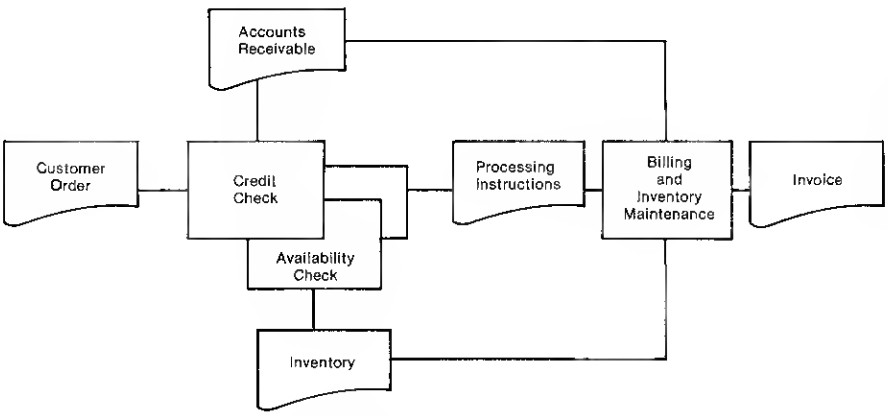
This system allows open item and balance forward accounts to exist in the same file. Open item systems retain the detail of unpaid invoices until they are paid or written off. A balance forward system retains invoice detail until a statement is prepared (then all detail is purged).

A balance forward system works best if the number of outstanding invoices can be kept to a minimum. However, if many questions of cash application are likely when a payment is made then an open item system should be used.

One plus to the COBOL accounts receivable system is that it accommodates both types of receivable recordkeeping. This recordkeeping is usually part of the order entry invoicing cycle of a firm. Fig. 1 indicates, in a simplified way, processing steps and major systems affected. Invoicing, inventory maintenance and accounts receivable updating are performed at the same time.

At billing time, invoice detail information is usually obtained from an Inventory file. Because of this it is not unusual to integrate billing and inventory maintenance. Often the billing is also tied into order entry and warehouse control. Since the receivable file is only updated with the invoice total, a much lower volume of transactions is encountered. As a result, receivable systems usually do not include an invoicing module.

The COBOL receivable system is an exception. It allows you to generate an invoice or enter post billing information. If you are billing as well, invoice information must be entered line by line. Up to 99 lines are allowed. You can enter a general ledger account code, part number, quantity order, quantity shipped, description and unit cost.



Generalized Order—Invoice Cycle

Price extensions, sales taxes and invoice totals are automatically calculated. The system even distinguishes between taxable and non-taxable sales.

A customer's address is extracted from the accounts receivable file. If the shipping address differs from the billing address, the new address is entered. You can also enter an order date, ship-via-description, discount terms and a customer purchase order number.

All data entered is buffered on the disk. Data in the buffer can be printed for analysis or altered. When invoicing, the forms can be placed in the printer and printed all at once. Once data are printed, the program automatically transfers the invoice total to the sales entry file. Thereafter, you can alter these only by using the sales entry file maintenance routines.

Invoicing Module

The invoicing module is well done and unless it is used, the distribution of sales data to various general ledger sales accounts will be limited. The post invoicing module appears to accommodate only one general ledger sales account. Thus, if multiple sales accounts are affected by billing, multiple invoices will be required.

Other receivable transactions are accumulated in buffers before updating customer accounts. If unposted transactions are in the system, an account inquiry may not provide the most up-to-date information. For this reason Radio Shack recommends that you post transactions daily. As is their practice, a posting report is printed when accounts are updated.

On request, an account can be queried to determine status and open item information. The status report can also be printed.

The label printing program allows the accounts receivable file to be used as the source of names when mailing for promotional purposes.

Another available report is the system definitions that can assist you in coding new information.

It is often desirable to screen the accounts receivable file during a month. The account listing program allows you to have either a complete account list, accounts overdue or current accounts. Accounts can be printed in alphabetic or numeric sequence.

At the end of an accounting period an "end of period" menu is presented. The menu allows you to print the trial balance with or without updating. An update prints a detailed report showing each account, the new balance, all of the transactions for the period, and the finance charges that have been applied. This detail should be retained as a permanent record.

The next step in the end of the period processing is to prepare customer statements. These are numbered so that if for any reason it becomes necessary to rerun some of the statements, the entire process need not be redone.

At the end of the statement printing routine, the account summary for a GL journal posting is prepared. After this is compiled, a GL distribution report is run. This terminates the monthly closing procedures for unintegrated systems. If you are using an integrated system, the GL transfer program can be run to transfer the data to the general ledger file.

Although this accounts receivable system represents a good approach to deal with billing and posting, Radio Shack has recently released a new version to handle even more functions. With this version, the three disk accounts receivable system is the first module of new COBOL Business Accounting Package. We will be looking at this new system in more detail in the future.■

EDUCATION 80

by Earl R. Savage

"Summer is around the corner, and the end of the school year is upon us."

Summer is around the corner, and the end of the school year is upon us. That means you will have more time for all those tasks you had to set aside during the academic year.

Well, summer is here and summer is later; it's time to get started. If you go about it more efficiently, you can get a lot done before fall and new classes are upon you. Here are a few suggestions to that end.

First, take a look at those document writing tasks. Study guides, grant proposals, etc. require writing and re-writing through several drafts. This is certainly an area in which your computer can be of great service. You can write the first draft and make as many changes as necessary without having to re-type the whole document. You can then make as many hardcopies as are needed.

To realize this efficiency you will need a word processor program of some kind. Many word processors are available commercially for a variety of prices.

The May, 1980, issue of *80 Microcomputing* published a comprehensive word processor on page 50. I used this program for some time and found it to be quite good. With a few minor changes to suit my equipment and needs, it turned out several hundred pages of manuscript with a minimum of effort.

What about increasing your program writing efficiency? There are myriad techniques and procedures for reducing your time and effort, including using a standardized subroutine package, merging program parts secured from various sources, and the use of "authoring" programs.

The most useful assistant you can have is a well chosen utility. I rarely sit down at my computer without first loading a utility which includes the functions I use most frequently. I'll give you a brief description of what mine does for me.

My utility includes debounce, reversible upper/lowercase driver, and auto-repeat on all keys. It has completely programmable single stroke keywords and macrokey with special repeat capability and pauses for fill-ins. Most importantly, there is screen oriented editing with such capabilities as duplicating, renumbering, moving lines and quickly creating multiple statement lines.

The total program is self-relocating and self-protecting, and can be enabled and disabled in whole or in part at any time from the keyboard. Does all that sound good? It is Omni-Key from Discovery Bay Software.

A good multiple function utility program can do wonders for you. It takes care of the mechanics of writing so that you can concentrate on developing your program.

Please add one more item to your to-do-list—share your work with other readers. Let me know what you are doing and how you are doing it. Send a sample if that helps to explain things. As space permits, I'll tell others about your accomplishments, which will encourage the rest of us to get busy and give us ideas for our own lists.

"The most useful assistant you can have is a well chosen utility."

Send your letters, notes, samples and whatever in hard copy, standard cassette, Exatron ES/F wafer, or TC-8 cassette with any of the three at normal speed or at the plus 50 percent speed option. Recorded material can be based on the Hindrichs' Word Processor or on Scripsit. On all tapes, be sure to label the system used, the speed and the word processor, if any.

CAI Planning Mechanics

Last month we discussed aspects of making a plan for a CAI (Computer Assisted Instruction) lesson and turning it into a CAI program. Now I would like to pass along to you a handy approach to the mechanics of planning.

You can waste a lot of time when planning. I've seen folk re-write entire sections of a plan just to insert something that had been overlooked or to change a point or two. You need a writing system which provides maximum flexibility for the inevitable changes which take place.

A plan is not the end product of CAI: It is nothing more than a plan for the instructional program—a guide. Once the program is completed, the plan is useless. For that reason, the plan should not be prepared in a polished, finished form.

Maximum flexibility can be achieved with a stack of unruled file cards. A single line across the narrow dimension will give you two areas on each card.

Holding the card vertically, think of the top half as the face of the computer display screen. Write in this area what you want the student to see on the screen at a given point in your program. Use the lower half of each card for notes about the program statements to be written later.

Changes in the instructional sequence are simple to handle: Just change the order of the cards. Add and delete by inserting or pulling out cards.

Seeing facsimiles of what the student will see on the display gives you a very good "feel" for the program. You can thus more easily spot weaknesses in the plan before the first program statement is written. Fewer changes will be required in the program writing stage.

Sometimes we become so involved in an undertaking that we cannot see it objectively. Have a knowledgeable friend look over your plan for things you may have missed. A pack of cards will provide a better idea of the finished product than a series of written descriptions.

Late Flash

I just received an interesting letter from Don Willard in Illinois. Don has arranged with the powers that be to have access to the school computer lab during off hours this summer. Several of his more accomplished students have volunteered to give a hand with some of the programs on Don's summer list.

Students who have TRS-80's at home will do the bulk of the writing there. They will meet with Don at the school lab from time to time for critiques of their work and to get new assignments.

Don sounds like a guy who knows how to spread the work around. He should get a lot accomplished this summer. ■

80 APPLICATIONS

by Dennis Kitsz

"Paper tape may no longer be the program storage medium it once was... but it still has its place."

You have the chance to pick up some terrific programs written for an 8080 based computer. The prices are so low it's worth checking them out. When they arrive you find . . . a dozen rolls of paper tape!

Paper tape may no longer be the popular program storage medium it once was, but for archival storage or communicating among different styles and types of computers, it still has its place. This article will describe a hardware and software interface between the Raeco TPR-1 paper tape reader and the Model I TRS-80.

The TPR-1 reader is sold by Raeco (Box 165, Washington, ME 04574.) The unit consists of a honed aluminum track for the paper tape, and a circuit board attached to the track. On the board are two integrated circuits, nine light sensors, an LED test light, resistors, and a 14-pin DIP socket. It is sold with a comprehensive technical manual for $32.50.

Tape Date Storage

Eight-level paper tape is capable of storing parallel bytes of data by means of holes punched in the tape. A smaller, ninth hole is used to provide a timing signal for the reading program. The ninth hole can also be used as a data-ready signal because, by the time the light triggers along the edge of the smaller hole, the larger holes are letting in enough light to be stable.

The TPR-1 reader is already set up for a computer bus, and hence is ideal for interfacing with the TRS-80. Its output is in parallel, and all signals are three-state. Because it uses only 12 mA, the reader can be run from the TRS power supply.

Fig. 1 is the diagram of the TPR-1. CMOS integrated circuits U1 and U2 evaluate the state of the light sensitive transistors and provide a parallel output. Part of U2 is used to drive the LED, which lights up whenever data is stable at the output of the reader.

Fig. 2 is the TRS-80 interface. Z1 and Z2 decode the port address 3F hex (63 decimal) in order to activate three-state buffer Z3. Ideally, the TPR-1 would have been designed with the Ready line separately activated from the data lines. That way, Ready

Program Listing

```
                 ; TAPE READER AND 256-BYTE LOADER ROUTINE
                 ; DENNIS BATHORY KITSZ
                 ; ROXBURY, VERMONT  Ø5669
                 ;
4Ø13             VECTOR  EQU     4Ø13H
Ø6CC             BASIC   EQU     Ø6CCH
3CØØ             VIDEO   EQU     3CØØH
                 ;
                 ; INTERRUPT VECTOR AT 4Ø12H
4Ø12                     ORG     4Ø12H
4Ø12 C3                  DEFB    ØC3H
7DØØ                     ORG     Ø7DØØH
                 ; CLEAR SCREEN SUBROUTINE
7DØØ 21ØØ3C      CLEAR   LD      HL,VIDEO
7DØ3 11Ø13C              LD      DE,VIDEO+1
7DØ6 Ø1FFØ3              LD      BC,Ø3FFH
7DØ9 362Ø                LD      (HL),2ØH
7DØB EDBØ                LDIR
7DØD C9                  RET
                 ; SCAN FOR ENTER SUBROUTINE
7DØE 3A4Ø38      ENTER   LD      A,(384ØH)
7D11 FEØ2                CP      2
7D13 2ØF9                JR      NZ,ENTER
7D15 C9                  RET
                 ; DISPLAY MESSAGE SUBROUTINE
7D16 7E          DISPLY  LD      A,(HL)
7D17 A7                  AND     A
7D18 C8                  RET     Z
7D19 12                  LD      (DE),A
7D1A 23                  INC     HL
7D1B 13                  INC     DE
7D1C 18F8                JR      DISPLY
                 ; CONVERT TO ASCII SUBROUTINE
7D1E F5          CONVRT  PUSH    AF
7D1F E6FØ                AND     ØFØH
7D21 1F                  RRA
7D22 1F                  RRA
7D23 1F                  RRA
7D24 1F                  RRA
7D25 FEØA                CP      ØAH
7D27 3ØØ4                JR      NC,HIBYTE
7D29 C63Ø                ADD     A,3ØH
7D2B 18Ø2                JR      NEXT
7D2D C637        HIBYTE  ADD     A,37H
7D2F 77          NEXT    LD      (HL),A
7D3Ø 23                  INC     HL
7D31 F1                  POP     AF
7D32 E6ØF                AND     ØFH
7D34 FEØA                CP      ØAH
7D36 3ØØ4                JR      NC,HIBTE2
7D38 C63Ø                ADD     A,3ØH
7D3A 18Ø2                JR      NEXT2
7D3C C637        HIBTE2  ADD     A,37H
7D3E 77          NEXT2   LD      (HL),A
7D3F C9                  RET
                 ; CLEAR TAPE READER INTERRUPT ACK.
7D4Ø F3          SERVEØ  DI
7D41 AF                  XOR     A
7D42 C9                  RET
                 ; PAGE ADDRESS INTERRUPT SERVICE
7D43 F3          SERVE1  DI
7D44 DB3F                IN      A,(3FH)
7D46 CD1E7D              CALL    CONVRT
7D49 AF                  XOR     A
7D4A C9                  RET
                 ; READ DATA / PLACE ON SCREEN INTERRUPT
7D4B F3          SERVE2  DI
7D4C DB3F                IN      A,(3FH)
7D4E 77                  LD      (HL),A
7D4F 81                  ADD     A,C
7D5Ø 4F                  LD      C,A
7D51 23                  INC     HL
7D52 AF                  XOR     A
7D53 C9                  RET
                 ; CHECKSUM INTERRUPT ROUTINE
7D54 F3          SERVE3  DI
7D55 DB3F                IN      A,(3FH)
```

Photo 1. The Final Product

could be tested at all times, whereas data would only be input whenever Ready indicated stable data. In its present configuration, a separate buffer must be used for the TPR-1 data lines.

Z4 is a flip-flop which sends an interrupt signal to the TRS-80 INT line; INTAK (interrupt acknowledge) is used to clear the interface flip-flop once the data have been read.

The entire circuit can be wire-wrapped on a small piece of perfboard, and mounted inside the case with the TPR-1. A detachable 40-pin cable is used in the prototype.

Page Read Software

Listing 1 presents the software to read a page (256 bytes) of data into the TRS-80 and store it in memory. At power-up, the interrupt patch point at 4012 hex is initialized with C9, a RETurn instruction. In its place, a patch is made to one of three interrupt service routines which will read

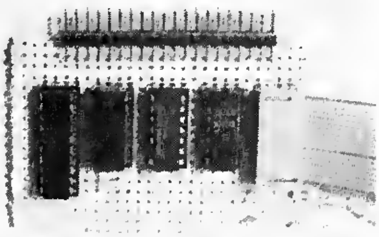


Photo 2. The Inner Works

```
7D57 47                      LD      B,A
7D58 AF                      XOR     A
7D59 C9                      RET
                   ; MESSAGES FOLLOW
7D5A 54            MSGNO1    DEFM    'THREAD TAPE AND PRESS CLEAR.'
7D76 00                      DEFB    00
7D77 4C            MSGNO2    DEFM    'LOADING PAGE ADDRESS:  '
7D8E 00                      DEFB    00
7D8F 42            MSGNO3    DEFM    'BYTES LOADING AS FOLLOWS:'
7DA8 00                      DEFB    00
7DA9 43            MSGNO4    DEFM    'CALCULATED CHECKSUM IS:  '
7DC2 00                      DEFB    00
7DC3 43            MSGNO5    DEFM    'CHECKSUM AS READ IS:  '
7DD9 00                      DEFB    00
7DDA 43            MSGNO6    DEFM    'CHECKSUM ERROR IN THIS BLOCK.'
7DF7 00                      DEFB    00
7DF8 42            MSGNO7    DEFM    'BLOCK LOADED CORRECTLY.'
7E0F 00                      DEFB    00
7E10 41            MSGNO8    DEFM    'ANOTHER BLOCK?  REPLY 1 FOR YES, 2 FOR NO'
7E39 00                      DEFB    00
7E3A 50            MSGNO9    DEFM    'PRESS CLEAR TO RETURN TO BASIC.'
7E59 00                      DEFB    00
                   ; REMEMBER THIS IS ENTRY POINT AND NOT!
                   ; BEGINNING OF PROGRAM.......
                   ; CLEAR SCREEN, DISPLAY "THREAD" MESSAGE
7E5A CD007D        START     CALL    CLEAR
7E5D 215A7D                  LD      HL,MSGNO1
7E60 11003C                  LD      DE,VIDEO
7E63 CD167D                  CALL    DISPLY
7E66 CD0E7D                  CALL    ENTER
                   ; DISPLAY "ADDRESS" MESSAGE & FIND IT
7E69 21777D                  LD      HL,MSGNO2
7E6C 11403C                  LD      DE,VIDEO+40H
7E6F CD167D                  CALL    DISPLY
7E72 21407D                  LD      HL,SERVE0
7E75 221340                  LD      (VECTOR),HL
7E78 37                      SCF
7E79 ED56                    IM      1
7E7B FB                      EI
7E7C 38FE                    JR      C,$
7E7E 21437D                  LD      HL,SERVE1
7E81 221340                  LD      (VECTOR),HL
7E84 21573C                  LD      HL,VIDEO+57H
7E87 37                      SCF
7E88 FB                      EI
7E89 38FE                    JR      C,$
                   ; DISPLAY "BYTES" MESSAGE & LOAD 256
7E8B 218F7D                  LD      HL,MSGNO3
7E8E 11803C                  LD      DE,VIDEO+80H
7E91 CD167D                  CALL    DISPLY
7E94 214B7D                  LD      HL,SERVE2
7E97 221340                  LD      (VECTOR),HL
7E9A 21003D                  LD      HL,VIDEO+100H
7E9D AF                      XOR     A
7E9E 4F                      LD      C,A
7E9F 0600                    LD      B,00H
7EA1 37            LOOP2     SCF
7EA2 FB                      EI
7EA3 38FE                    JR      C,$
7EA5 10FA                    DJNZ    LOOP2
                   ; DISPLAY "CHECKSUM CALC" MESSAGE
7EA7 21547D                  LD      HL,SERVE3
7EAA 221340                  LD      (VECTOR),HL
7EAD 21A97D                  LD      HL,MSGNO4
7EB0 11403E                  LD      DE,VIDEO+240H
7EB3 CD167D                  CALL    DISPLY
7EB6 79                      LD      A,C
7EB7 D5                      PUSH    DE
7EB8 E1                      POP     HL
7EB9 CD1E7D                  CALL    CONVRT
                   ; DISPLAY "CHECKSUM READ" MESSAGE & DO IT
7EBC 21C37D                  LD      HL,MSGNO5
7EBF 11803E                  LD      DE,VIDEO+280H
7EC2 CD167D                  CALL    DISPLY
7EC5 FB                      EI
7EC6 37                      SCF
7EC7 38FE                    JR      C,$
                   ; DISPLAY CHECKSUM AND CHECK IT
7EC9 78                      LD      A,B
7ECA D5                      PUSH    DE
7ECB E1                      POP     HL
7ECC CD1E7D                  CALL    CONVRT
7ECF 78                      LD      A,B
7ED0 B9                      CP      C
7ED1 280B                    JR      Z,CKSMOK
                   ; DISPLAY CHECKSUM BAD MESSAGE
7ED3 21DA7D                  LD      HL,MSGNO6
7ED6 11C03E                  LD      DE,VIDEO+2C0H
7ED9 CD167D                  CALL    DISPLY
7EDC 1809                    JR      LEAVE
                   ; DISPLAY CHECKSUM OKAY MESSAGE
7EDE 21F87D        CKSMOK    LD      HL,MSGNO7
7EE1 11C03E                  LD      DE,VIDEO+2C0H
7EE4 CD167D                  CALL    DISPLY
7EE7 21107E        LEAVE     LD      HL,MSGNO8
7EEA 11003F                  LD      DE,VIDEO+300H
7EED CD167D                  CALL    DISPLY
```

each byte of data as it becomes stable at the output of the TPR-1.

The program is entered at line 1160. The screen is cleared, and the user is prompted to enter a base address in hex. This is an address starting at which the tape data is to be loaded into memory. The keyboard is scanned for characters 0 to 9 and A to F; these are displayed, and when Enter is pressed, they are converted to a starting address.

```
                01840 ; SCAN KEYBOARD FOR 1 OR 0 & DO IT
7EF0 3A1038     01850 FINDYN  LD      A,(3810H)
7EF3 FE02       01860         CP      2
7EF5 CA5A7E     01870         JP      Z,START
7EF8 FE04       01880         CP      4
7EFA 2802       01890         JR      Z,DONE
7EFC 18F2       01900         JR      FINDYN
7EFE 213A7E     01910 DONE    LD      HL,MSGNO9
7F01 11403F     01920         LD      DE,VIDEO+340H
7F04 CD167D     01930         CALL    DISPLY
7F07 CD0E7D     01940 LOOKNG  CALL    ENTER
7F0A C3CC06     01950         JP      BASIC
7E5A            01960         END     START
00000 TOTAL ERRORS
```

"When they arrive you find a dozen rolls of paper tape!"

The tape must then be threaded. If it is threaded after data reading has begun, false initial information will be presented to the TRS-80 (line 1680). The tape reading is begun at line 1740.

With this software, the tape to be read must be in the following format:

- One byte code of information (tape number, address page, etc.), which is displayed but not stored in memory.
- 256 bytes of data.
- Simple one byte checksum.

If the tape is not in this format, the program can be easily altered to accommodate any other data block format.

Interrupts are then enabled (lines 1810-1820), and a series of short interrupt service routines are activated. The first routine merely waits for the interrupt line to clear, as it may be set by stray light in the room when the tape is threaded (lines 1820-1870). Two hundred and fifty six bytes are then loaded and displayed (lines 1910-2030). The checksum is calculated and displayed (lines 2050-2130), and the checksum is read from tape and displayed (lines 2150-2280). If there is a match, the memory pointer is advanced in order to read the next block of tape; otherwise, it is reset to the beginning of the block, allowing the tape to be read again (lines 2210-2420). Finally, the option of loading additional blocks or returning to BASIC is presented (lines 2440-2540).

Using the TPR-1, the interface, and this simple software, 8080 programs as well as programs saved in an archival paper tape format may be read into your TRS-80. ■

Eds. note: Dennis flew out to the April computer show in San Francisco, and, blinded by the bright California sun and seduced by the warm air (it's still winter in Roxbury, VT), forgot the way home. Some very specific and cunning threats have elicited a promise from him to return in time to present his regular column next month.

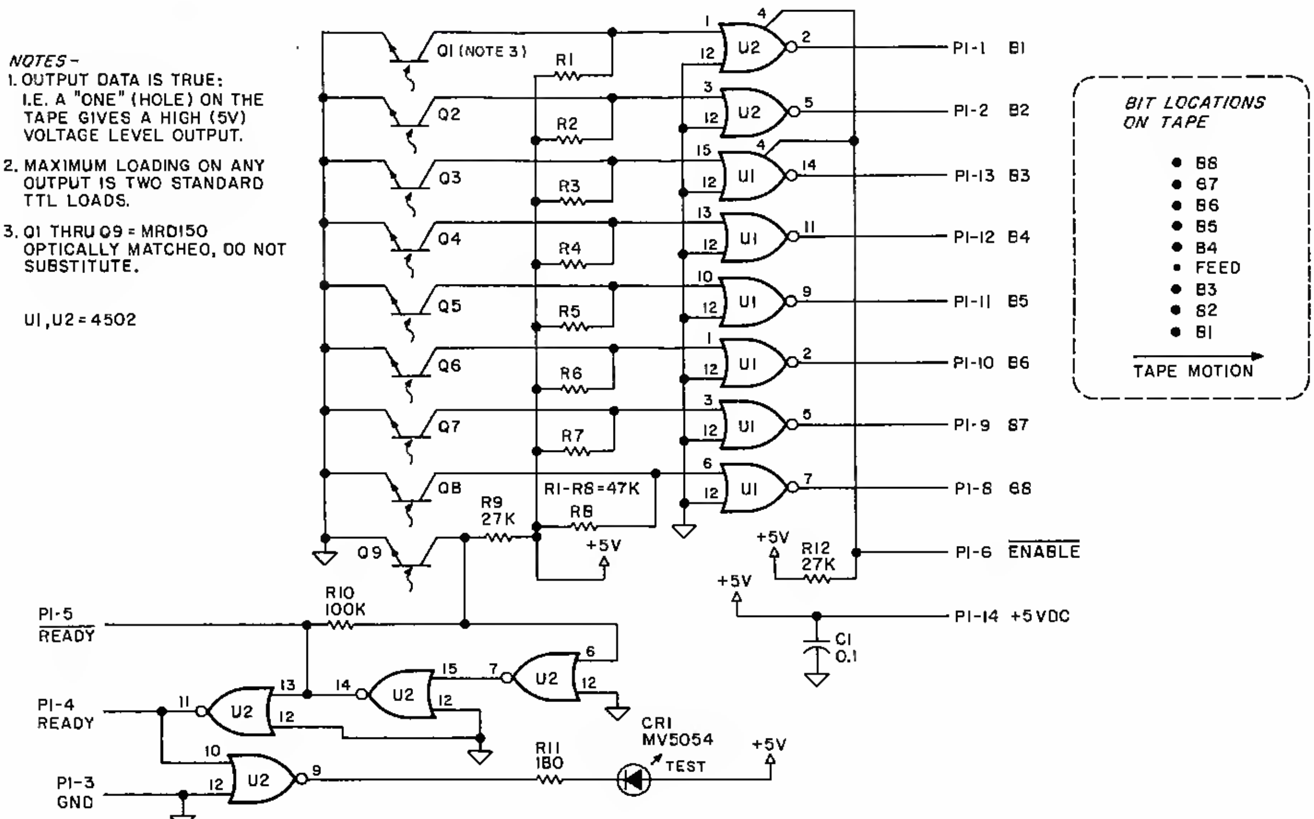
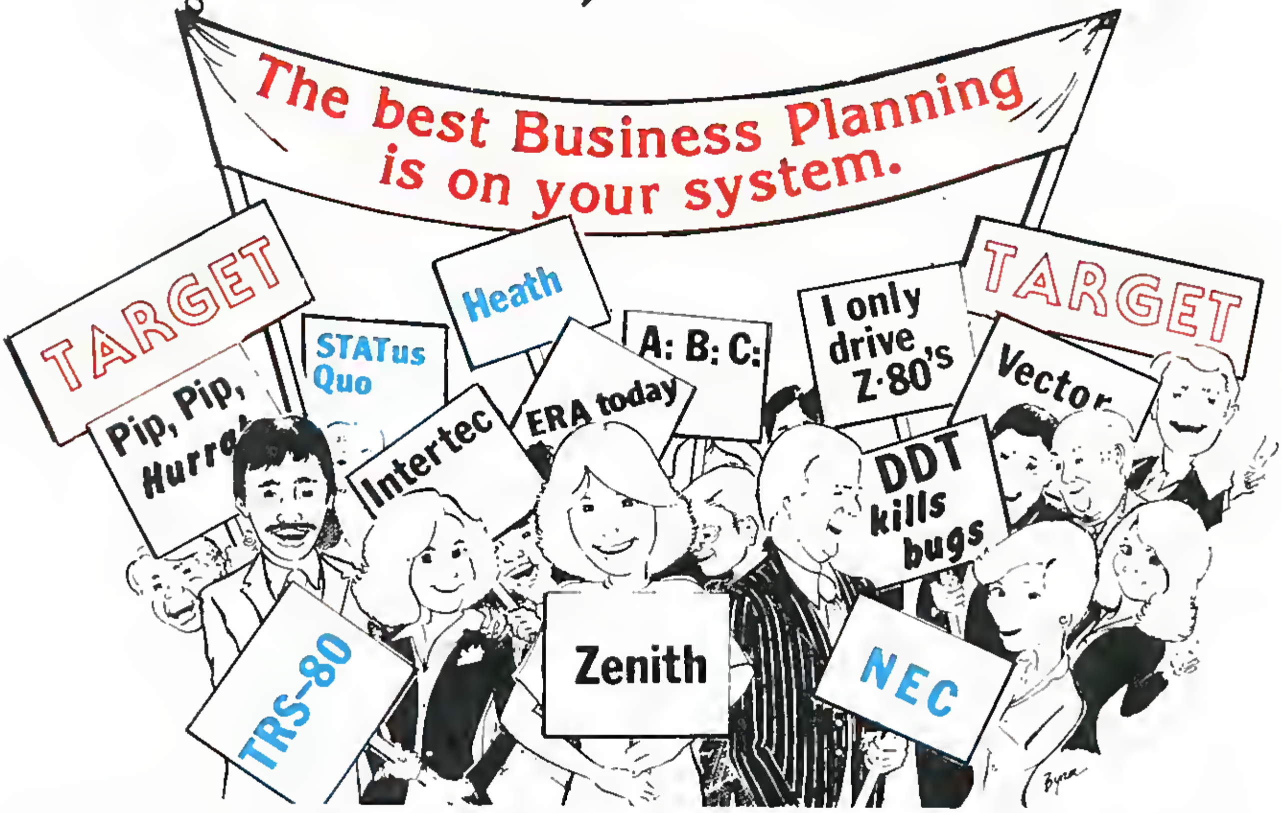


Figure 1

STAND UP, CP/M USERS



For the past few months you have patiently endured the indignity of watching your friends show off their flashy visible number cruncher on their game-playing computer and longed for something as slick.

You seriously considered buying *their* computer, but you just couldn't give up the benefits of the CP/M operating system.

Well, say hello to TARGET™, the best business planning system available on micros today (and some big computers besides).

Talk about a blank ledger sheet or columnar tabulator! TARGET™ is much more than that. Sure, just like their product, you can fill in the rows and columns as you see fit and the numbers will whiz by you on the screen. But, we let you create your report in English, not in some combination of reverse Polish notation and algebraic matrix languages.

Our TARGET on *Your* System. . .

```
LINE 1 SALES  = 100 200 300 400
LINE 2 EXP    = GROW 50 BY 15%
LINE 3 NET    = SALES  -  EXP
```

Their Product on *Their* System . . .

```
SALES 100 200 300 400
EXP 50  +B2*1.15  +C2*1  +D2*1.15
NET  +B1-B2  +C1-C2  +D1-D2  +E1-E2
```

At least, that is what their product might look like if you could see all of your data and calculation rules at the same time, *which you can't*. If you think that it is an easy approach for debugging, guess again.

TARGET™ displays a full screen of results or data and calculation rules at your command. And, it runs on your system. How much more could you ask?

Lots! And TARGET™ delivers. We give you the ability to:

- Obtain hard copy printouts of *both* the results *and* your set of data and calculation rules used to create those results.
- Write the results of data and calculation rules to the disk as formatted files for word processing.
- Use full conditionals (IF . . . THEN . . . ELSE) in your set of rules and stack conditionals within conditionals.
- Build powerful models with commands such as GROW, MIN, MAX, AVE, CUM, GREATER, LESSER, SUM.
- Edit lines with ease.
- Enter rules and data in any order you wish. TARGET™ will automatically sort rows in ascending order *and* process your model correctly even if LINE 33 SALARIES needs to be calculated before LINE 3 EXPENSES.

There is so much more about the TARGET™ Planner that will impress you. Our manual explains not just what a command or function is, but how to use the functions in everyday business situations. Our manual and our newsletter give you illustrative examples of business planning problems and how TARGET™ can help you solve those problems. For example, we will show you how to quickly determine the payback period for a proposed project.

TARGET™ is a compiled system running under CP/M. There is no need to buy BASIC or FORTRAN or any other system software. And the price is *only $195* for the TARGET™ Planner, our basic system.

Stand up and be proud, CP/M users. The best business planning system runs on the best operating system on *your* computer.

Advanced Management Strategies, Inc.
1935 Cliff Valley Way, N.E., Suite 200
Atlanta, Georgia 30329
404/634-9535

260

P.S. When all of your friends start drooling over *your* product, tell them to cheer up. With Microsoft's great SoftCard and 16-K memory board, your friend's Apple can move up to your operating system and run TARGET™.

CP/M is a registered trademark of Digital Research.
SoftCard is a trademark of Microsoft.
TRS-80 is a trademark of Tandy Corporation.

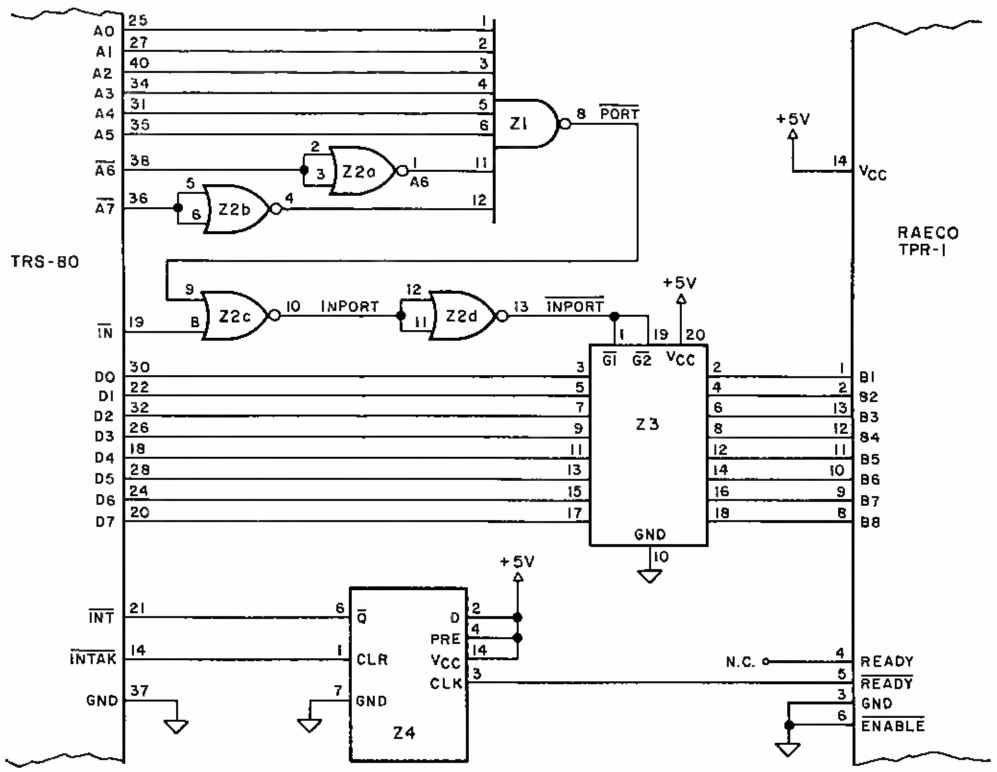


Figure 2

80 CALENDAR

June

The National Forum on Computers and Health, Alexandria, VA is offering a workshop called **Minicomputers and Microprocessors in Medical Practice Management.** It will be held **June 4-5** in Philadelphia. To register call (703) 549-8020 collect. The toll free number for Virginia residents is (800) 336-4776.

June 30 is the deadline for submissions for **The Johns Hopkins First National Search** for personal computing devices, programs and designs to aid the handicapped. The search is designed to "discover existing applications and to inspire new ideas for the application of personal computing to meet the needs of the handicapped," according to a press release from Johns Hopkins University.

A $10,000 grand prize is offered along with 100 other awards. The three submission categories are defined as 1) Computer Based Devices which includes "hardware invented or modified for the purpose, or working hardware and software which can demonstrate a new application," 2) Computer Programs which means "specialized software and concepts for existing computers," and 3) System Concept/Design, which is "written descriptions of ideas not yet implemented."

The competition is being sponsored by grants from The National Science Foundation and by Radio Shack.

For more information, write Personal Computing to Aid the Handicapped, The Johns Hopkins University, P.O. Box 670, Laurel, MD 20810.

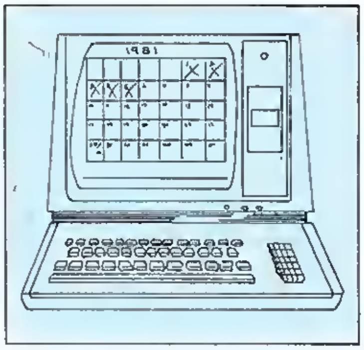
July

July 5-31 The Hill School, Pottstown, PA, will conduct **four one-week computer workshops** using the school's PDP 11/34 system and will offer students maximum hands-on experience. The first three workshops will be open to students of Grades 7-12. The last will be for teachers and other professionals.

Contact John E. Parnell, The Hill School, Pottstown, PA 19464, for information.

July 13-14 will see a **seminar on using the OASIS operation systems on Z-80 microcomputers** at Phase One Systems, Oakland, CA.

Classes will be limited to 20-30 students with plenty of "hands-on" activities. Price is $195. Information is available from Phase One Systems, 7700 Edgewater Dr., Suite 830, Oakland, CA, 415-562-8085.



Throw It Away

(your dictionary, that is)

Now you can proofread and correct ten pages of text in less than one minute, with **MICROPROOF** dictionary software

- **EASY TO USE:** Prepare your text on any Z-80 based microcomputer, using any of a number of popular word processing programs. When you are finished, enter the appropriate command, and MICROPROOF proofreads your document, displaying misspellings and typos on the screen. Then MICROPROOF displays each error separately, requesting you to enter the correct spelling for each. Finally, MICROPROOF corrects your document. All in less than a minute.
- **COMPREHENSIVE:** MICROPROOF comes with a 50,000 word vocabulary. That's equivalent to a Webster's Pocket Dictionary. And MICROPROOF's dictionary is INFINITELY EXPANDABLE. You can add your own technical words and jargon.
- **RECOGNIZES** prefixes, suffixes, hyphenation and comment lines.
- **REQUIRES** ONLY 32K of memory and one single or double density 5¼ inch or 8 inch disk drive.
- **AFFORDABLE:** Available in three forms: fully independent program to identify errors, independent program to identify and correct errors, or with a conversion program that will permit MICROPROOF to operate from within your own word processing software.

PRICES

TRS-80® Model I version	$125.00	Correcting Feature	$60.00
TRS-80® Model II version	$200.00	Word Processing Conversion	
CP/M® version	$200.00	**Scripsit® or Electric Pencil®**	$35.00
APPLE® version	available soon	Separate User's Manual	$5.00

Master Charge and Visa accepted. No COD please. California residents add 6% sales tax.
Send for a brochure—specify type of microcomputer and word processing software used. Dealer inquiries welcome.

CORNUCOPIA SOFTWARE 378
Post Office Box 5028
Walnut Creek, California 94596
(415) 524-8098

Let MICROPROOF Find Your Errors

Before Someone Else Does!

80 NEWS

edited by Bert Latamore

"Our feeling is this is a powerful new medium. If we are serious about electronic publishing, there has to be a way to design editorial products for that medium."

Dow Forms Interactive News Desk

What may be the world's first electronic newsroom is being created for the TRS-80 and videotex user as part of the Interactive News Division of Dow Jones Co., Inc., Princeton, NJ, publisher of *The Wall Street Journal.*

The seven-man newsroom's aim, according to its editorial director, Richard J. Levine, is to "create a new product by seeing and showing relationships" among businesses and whole industries. They will reorganize information from existing Dow news publications, including *The Journal, Barron's National Business and Financial Weekly,* and Dow's financial news wire. They will supplement this with statistics and information from other published private and government records.

Their aim is to take advantage of the unique interactive nature of computerized news to give each user a large amount of information on the subject he asks about, Levine said.

Levine, *The Journal's* former chief economic writer, called the newsroom a major upgrading of a service previously concerned solely with encoding Dow wire stories for computer retrieval by professional financial people and Apple and TRS-80 microcomputer users.

The electronic service works by providing each user with information on subjects he defines by specifying the database he wants (news, stock quotes or financial statistics, for instance) and the specific subject area (individual company or group of industries.)

Levine's assignment is to remake the news database, which currently contains stories off the Dow Jones financial news wire from the last 90 days. He sees the interactive nature of the computer news service as freeing the newsman from the confines of space which limit his stories severely on a daily newspaper. And, since electronic news is user specific, it does not burden each reader with material on subjects he is not interested in.

Richard J. Levine

Two Major Moves

The newsroom evolved as the result of two major moves by Dow in the last two years.

In May 1979 Dow bought full control of its interactive information service from its partner in the venture, Bunker Ramo Corp., Oak Ridge, IL, a producer of information systems for brokerage houses and banks. A year later Dow reorganized into seven operating groups, one of them the Interactive News Division.

The moves signaled an overall expansion of Dow's electronic publishing.

Besides creating the newsroom, Dow recently arranged to make stock quotation and statistical information databases available on TRS-80 microcomputers and Videotex terminals.

Levine is also considering adding to the databases by using material from Dow's many other publications. They include a foreign news wire that reaches 40 countries, an Asian edition of *The Journal*, the Ottaway Newspaper Chain with daily papers in 20 cities, a textbook publisher and *Book Digest* magazine.

For instance, Levine said, little of the statistical information from *Barron's* is being put into the computer. He may have a new database created to contain the rest of *Barron's*.

Dow, which has been in electronic publishing since 1971, sees it as a "powerful new medium," according to Levine, who said the recent Dow moves are part of a decade-long evolution.

When Dow and Bunker Ramo went into electronic publishing, Levine said, they aimed at major computerized financial institutions and Fortune 500 companies. As smaller firms computerized, the service was marketed to them.

When microcomputers became widespread, Dow solidified an agreement with Apple Computer Inc. Cupertino, CA, in 1978 that made some databases available to users on a timesharing basis. A similar agreement this year with Tandy/Radio Shack, Fort Worth, TX, is a major expansion into that same market, said Levine.

The newsroom, he said, is a natural outgrowth of that outlook.

"When we get serious about something, we tend to involve journalists," Levine said.

Selecting Writers

To meet the challenge of creating an editorial product for this medium, Levine is choosing writers with a strong background in hard news. Like Levine, himself, they may not have great technical knowledge.

Levine was a *Journal* reporter for 14 years, serving as military correspondent, labor reporter and editor and White House and Congressional reporter. He became chief economic writer in 1976 and has written a front-page weekly column, "The Outlook," since 1977.

His deputy editor is Peter J. Schuyten,

former technology editor for *The New York Times.*

"Schuyten understands the technology, which I don't," Levine said. "But at heart he's a jounalist."

Electronic news cannot replace a newspaper's ability to deliver stories of general interest to large numbers of readers in a conveniently usable form, he said. For instance, the recent attempted assassination of President Reagan is basically a newspaper story, not a computer news subject, he said.

"We have a role, and I think that role is delivering specific information packages," he said.

Though electronic news will only supplement *The Journal*, not replace it, Levine said, it is possible that in the future a subscription to *The Journal* will include both the paper and electronic information.

The electronic newspaper will cover essentially the same areas *The Journal* does and, though it is offered to a different market, Levine is not planning to change his news policy to fit the specific non-business interests of that audience.

"Our intent is to build on our strengths in the business, financial and economic areas we know best," he said. "I don't want to detract from our basic business, which is to serve the businessperson."

"We're a publisher," Levine said. "We have news and information to sell. To the extent that we can, we will deliver it as widely as possible through whatever media are available." ■

by Bert Latamore
80 *Staff*

FCC Liberalizes Ruling

In a decision that will lighten the hearts of peripheral manufacturers everywhere, the Federal Communications Commission (FCC) has made some major concessions regarding certification of computer peripherals.

The commission has re-evaluated certain sections of the rules (15.843 (a)(3), 15.4 (s)) covering peripheral equipment compatible with Class B computers—those used in residential settings where radio frequency interference is most likely to be a problem.

Up to now, any peripheral that was capable of being connected to a Class B computer had to meet the same stringent certification requirements that the computer, itself, faced. In the future, peripheral manufacturers need not be constrained by the strict Class B limits if they can show the commission that the likelihood of their peripheral being used with a Class B computer in a residential/hobbyist application is small. One example is a manufacturer who might produce an extremely expensive printer costing many times the price of a popular microcomputer with which it could be used.

This change in attitude toward peripheral manufacturers marks a new realization by the FCC that the stringent RFI requirements which it has placed on a hitherto unregulated industry are causing economic hardship for many. To ease these "growing pains," as the commission terms them, the FCC has established a computing device panel to answer industry inquiries and to continually review regulations.

This panel, headed by the commission's chief scientist, provides peripheral manufacturers with an opportunity to plead their case. A unique aspect of the panel is the power provided the chief scientist to make immediate decisions, thus speeding up the usually sluggish bureaucratic decision-making process. The commission cites the Administrative Services Act in defense of its right to dispense with normal rule making procedures. Under this act, rule changes which are merely "interpretive," rather than substantive, may be made without the normal 30 day waiting period.

The net result of the FCC's change in rule interpretation is simply this: The manufacturer's marketing of a peripheral is the prime, but not the only, consideration in establishing the device's certification requirements as far as the FCC is concerned. A manufacturer may now petition the commission to reclassify his peripheral if it appears unlikely that the device will be used in residential or hobbyist applications. ■

by Chris Brown
80 *Staff*

✓See List of Advertisers on page 306

Sharp Selling RC Pocket Computer in US

The Japanese firm that makes the Pocket Computer for Tandy/Radio Shack has started marketing it in this country under its own name.

A spokesman for Tandy denied any link between Tandy's recent price cut of the Pocket Computer and the move by Sharp Electronics Corp., saying the sale was planned last fall and extra computers were ordered for it.

In fact, the spokesman, Ed Juge, director of computer merchandising for Tandy, Fort Worth, TX, said his company and Sharp were discussing the design of a second generation of the pocket computer.

Meanwhile, Don Lawrence, national sales manager of Sharp's Systems Division in Paramus, NJ, said they will market the Pocket Computer nationally through every outlet available.

He said they would be sold in department stores, computer specialty stores, discount stores and even college bookstores.

He said it would be sold as a scientific and business machine usable, for instance, in photographics darkroom work, computing airplane flight plans, and working with stocks and bonds.

The Sharp Pocket Computer was unveiled at the National Computer Convention in Chicago, IL, in early May, where Sharp also introduced its first microcomputer.

Lawrence said Sharp organized its Systems Division at its Paramus, NJ, headquarters to handle sales of all its programmable products. He confirmed that Tandy would continue selling the Pocket Computer and said from Sharp's standpoint Tandy is simply another major outlet for the product.

Juge declined to give any details of Tandy's "wish list" for the second generation Pocket Computer, saying, "We don't know what's feasible yet."

In an earlier statement, however, he said when Tandy decided to buy the Sharp product they had planned to have a completely redesigned version made as soon as possible. ■

Deathmaze Winner

Ah, readers—it is so good to sleep quietly at night, not dreaming of twisting, empty hallways and carefully stockpiled mounds of molding, useless rubbish. No more twilit horrors pouncing on me from corridors without direction signs. I have walked the halls of Deathmaze, and lived!

Technical Sergeant James R. Roth (U.S.A.F.) is the guide who led me through the maze safely. He called here early one morning (which must have been very early California time) and filled me in on the details. He later sent his maps and very involved instructions. He tells me that he had help with the solution from the Williams family. TSgt. Roth will be on his way to Spain by press time.

Many other readers called and wrote with answers, and I thank you all for your efforts. Some gamesters who hadn't even played the game sent in suggestions, hoping to help me out of my quandry. I hope you all enjoy the game as much as I did.

Debra Marshall
80 *Staff*

Tandy, Tokyo Firm Join In Japanese Market Rush

Tandy/Radio Shack and the Tokyo Electric Co. (TEC) are making a joint assault on the Japanese microcomputer market with an agreement that allows TEC to sell the TRS-80 Models I, II and III microcomputers in its new computer centers.

TEC will also manufacture TRS-80 Model I's for the Japanese market.

The agreement effectively combines the new TEC stores and the existing Japanese Radio Shack outlets into what Tandy President John V. Roach and TEC President Toshio Akitsu said would be the largest sales and service set-up in the Japanese microcomputer market.

It is a major bid by Tandy for a larger share of that market. Tandy has been successfully selling its TRS-80s through its own stores in Japan for three years according to Gerald P. Asher, director of financial planning for Tandy.

TEC is a Tokyo-based electronics manufacturer, a member of the Toshiba group of companies, and has recently set up a Small Business Computer Division to market microcomputers. The firm has opened a computer center in Tokyo and has centers in Osaka and Magoya, Japan, in the works. As part of the agreement the two firms will work together to extend their sales and service network throughout Japan.

Tandy and TEC have worked together in the past. The Tokyo firm has custom made several items for Radio Shack, including some of their best line printers, Asher said. TEC is the leading manufacturer of electronic cash registers and point of sale terminals in Japan.

A Good Value

Asher said the TRS-80 microcomputers are a success in Japan because they are a good value for their price. Stacked up against the majority of Japanese competition, he said, they tend to be less fancy but also less expensive.

"I don't think there's any question the Japanese are making good equipment," Asher said. Some of the Japanese equipment he has seen is better than the American equipment.

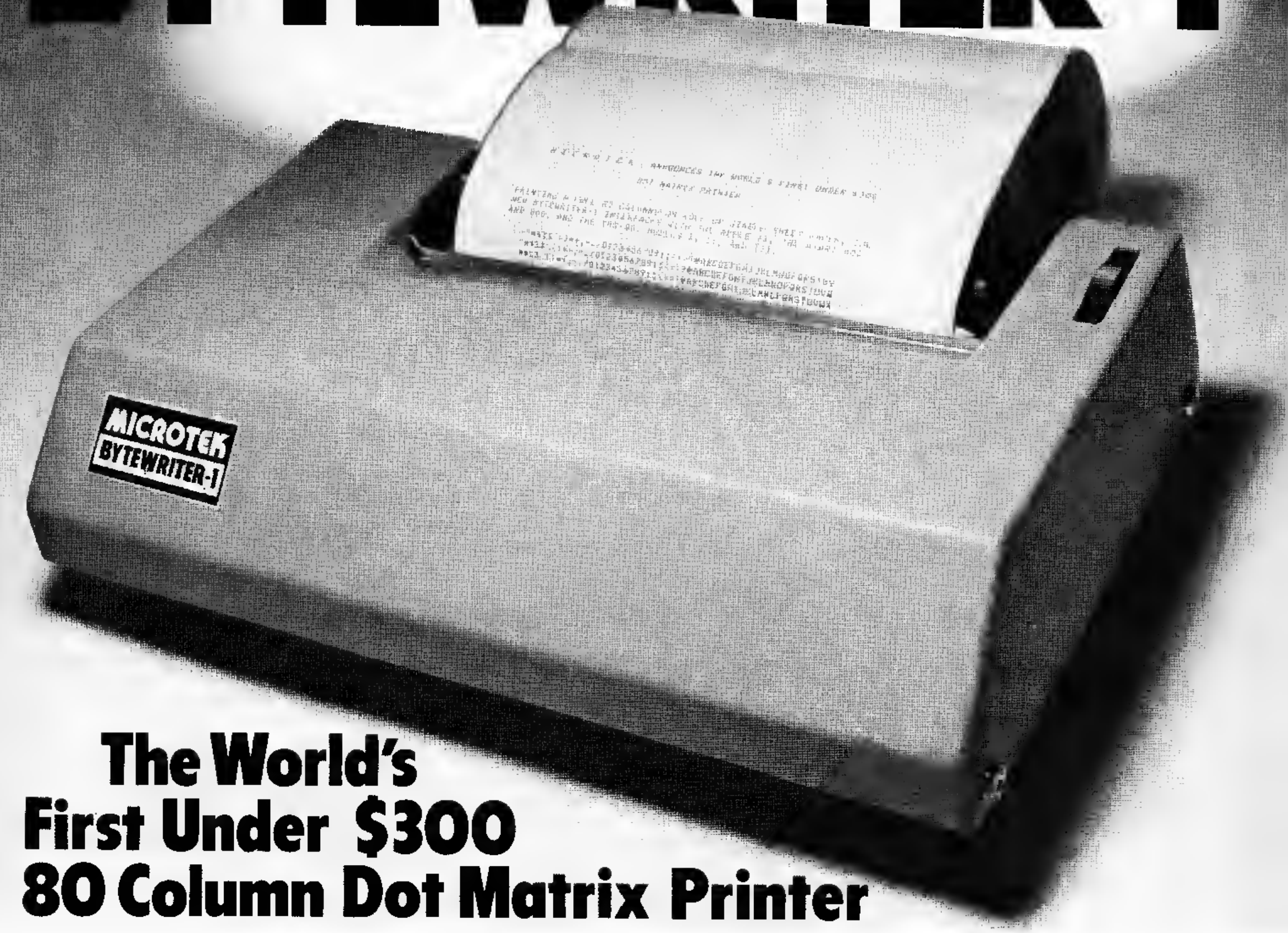
However, he said, "Most of the 'for real' equipment I've seen is in the $4,000 class. They're not low end products."

Asher said the Japanese machines do not represent any significant technological breakthroughs. There is nothing they do that the TRS-80s can't do. According to Asher, the Japanese products are simply refined further. But not everyone is interested in paying for this further refinement.

In fact, Asher said, the TRS-80 Model I, the most basic of the Radio Shack desktops, is still popular in Japan and Europe both. It is no longer sold in the US because of Radio Frequency Interference standards set by the Federal Communications Commission (FCC) for microcomputers.

Asher said adequate software packages are available for the TRS-80s in Japan. These products, he said, are all developed in Japan for that market. ■

BYTEWRITER-1



The World's First Under $300 80 Column Dot Matrix Printer

PRICE BREAKTHROUGH FROM MICROTEK

Bytewriter-1, the $299 printer that has taken the industry by storm! A product made possible through cost-effective design, efficient manufacturing operations, a proven low-cost print mechanism and the simplest possible control electronics. We have done extensive testing with all three models of the TRS-80, the Apple II, and the Atari 400 and 800. That is why we can guarantee you won't have any interface problems with the Bytewriter-1.

TRS-80 is a trademark of Radio Shack, Div. of Tandy Corp.
Apple II is a trademark of Apple Computer, Inc.
Atari 400 & 800 are trademarks of Atari, Inc.
Bytewriter-1 is a trademark of Microtek, Inc.

Microtek, Inc. ~360
9514 Chesapeake Drive
San Diego, CA 92123
Tel. 714-278-0633
Outside Calif. call
toll free: 800-854-1081
TWX. 910-335-1269

MICROTEK inc.

BYTEWRITER-1 SPECIFICATIONS

Printing Technology:	7-wire bi-directional impact wire matrix
Printing Speed:	60 lines per minute (80 characters per second) continuous
Character Set:	96 character ASCII (upper and lower case)
Character Size:	10 characters per inch (80 columns per line) plus expanded printing
Paper:	Friction feed (synchronous), accepts single sheet and roll paper up to 9½ inches maximum width. Prints original plus 3 copies.

By the way, our replacement print-head costs less than $30 too!
Call or write today for more information.

Shack's Hopes High for Hard Disk

A hard disk system will be available for Tandy's Model II microcomputer by the end of 1981 according to Ed Juge, director of computer marketing at Tandy/Radio Shack, Fort Worth, TX.

Juge said both the hardware and software phases of the project are on schedule, and he foresees no delays. He said, however, that no announcements will be made by Tandy until enough hard disk drives are warehoused around the country to satisfy the expected heavy demand.

The Model II hard disk will be approximately a 10 megabyte stand-alone device requiring no special interface or cabling. The disk operation system (DOS) is expected to be extremely advanced and business oriented.

Several Sources

Juge offered "no comment" when asked who the OEM (original equipment manufacturer) of the drives would be, but *80 Microcomputing* has learned that Tandy is considering several sources.

When queried about operating software, Juge said, "As far as I know, the DOS is being developed in-house, although portions may be contracted out to specialty software houses if necessary."

One source of hard disk drives that Radio Shack is considering is Micro Peripherals Inc., Oklahoma City, OK, a division of Control Data, Inc. Ken Nichols, manager of public relations for Control Data, said his organization has sent several Finch Drives (hard disk units) to Tandy for evaluation. The Finch Drive, manufactured by Micro Peripherals, is an eight-inch unit that has 24 Mbytes of storage and a data transfer rate of 6.4 bits. This Winchester-type unit is a sealed module and operates at 3600 rpm with a head clearance of 25 micro inches. It employs a voice coil linear motor, hard sectored disks and has built-in data recovery circuitry.

As supplied by MPI, it has no interface circuitry. Radio Shack must design the interface logic required by the Z-80 chip.

Nichols told *80 Microcomputing* that MPI will be shipping the Finch Drive to customers in quantity by June and added, "At this point it is up to Tandy as to when any hard disk system using our drives is available."

While Juge said that, for the most part, the DOS would be developed in-house, *80* has learned Tandy has been looking to outside consultants for help with the hard disk system. Pick and Associates of Irvine, CA, a small but experienced software development firm, is one of the organizations to which Tandy has turned.

Tim Holland, the senior VP of Pick, told *80 Microcomputing* that his organization has been involved in negotiations with Tandy, but not for an occasional software patch within TRSDOS. Instead, they have been looking at adapting the entire Pick operating system to the Model II. He added, "Tandy is looking for an extremely sophisticated, business-oriented system for the Model II. We feel that the Pick system fits the bill nicely."

The Pick system is already running in Z-8000 applications and on hardware manufactured by Honeywell, Hewlett-Packard, Evolution Systems, Microdata and Applied Digital Data Systems. Over 1,000 Pick operating systems have been sold at $1,000 a copy.

According to Holland, the Pick system has many benefits. These include extremely detailed documentation, optimal Z-80/Z-8000 operations, excellent data base management, a reportorial language called Access and full networking capability using DMA (direct memory access) techniques.

An added plus for the Pick system in Holland's view is the fact that it is already running with hard disk drives that Tandy is considering, those of ADDS (Applied Digital Data Systems). Holland optimistically told *80*, "At this point we feel that we've got the inside track."

A senior contract negotiator for Pick, who preferred to remain anonymous, was more reserved in his outlook. He said that his client had gambled heavily on Radio Shack choosing the Pick system and, consequently, much time and energy has been committed to preparing the DOS by January 1982. He did predict, "We will soon see a working relationship between Pick and Tandy."■

by Chris Brown
80 *Staff*

Juge Visits Wayne Green

Tandy/Radio Shack recently sent Director of Computer Merchandising Ed Juge and Consumer Information Officer Bill Walters, north from their Fort Worth, TX, headquarters to tour Wayne Green, Inc., Peterborough, NH, publishers of *80 Microcomputing*.

While here they held a two-hour news conference with the magazine's staff during which Juge revealed some of his personal thoughts and Tandy's official ones on topics ranging from the importance of microcomputing software to Radio Shack's continuing support for the Model I.

"Software is the future of the industry," Juge said.

The emphasis on hardware and "doing your own thing" will die down, he said, as more people in both business and the home buy microcomputers to serve as appliances.

These people, he said, will not be interested in learning to program. They will want to buy their software ready to use, directly applicable to their needs.

This market will grow so large, he said, it will support specialty stores which will sell only software, just as bookstores only sell books.

Specialized Software

It will also support highly specialized software houses, he said, which will produce programs for one small area, such as inventory control.

Such a house, he said, might base its work on two or three programs. They would constantly research their area and bring out many versions of their programs, each designed to fit the needs of a particular industry.

While increasing numbers of small businesses ranging from pharmaceutical houses to engineering firms are developing specialized programs to suit their uses, Juge threw cold water on the idea that these firms will turn to selling their software.

Often, he said, such a company will be better off keeping a program to themselves, especially one that solves a problem and gives them a competitive edge. If they do decide to market it, he said, they may find they have taken on more than they bargained for.

"We get all kinds of people who say they've developed a program that does a good job for them," Juge said. Unfortu-

nately, those programs are often designed so specifically that they cannot be used by anyone else.

For instance, he said, a small print company may write a program to estimate the cost of jobs. It may work well for them, but only handle jobs involving less than 100,000 copies because that shop only does work under that size.

If a print shop that does larger jobs buys the program, they will find it useless.

Juge said Tandy has between 80 and 100 programmers developing software for the TRS-80 computers. This spring they doubled the size of their systems software department in eight weeks and, "If we could find the people, we would double it again."

Tandy has the same problem in at least one other area, Juge said. He is constantly looking for business program reviewers. This small group must evaluate a program's worth to the small businessman. Though he has essentially an unlimited budget for the group, he cannot find enough people with both the technical knowledge and business experience necessary.

BASIC as Standard

Juge said BASIC will probably remain the standard microcomputer language for the foreseeable future. It combines utility with ease of use for the non-professional, he said.

He sees a need for new intermediate languages for complex programming applications, however. At present programmers often have to develop their own.

For instance, he said, Tandy recently had a programmer develop a COBOL for the TRS-80 for Tandy's private use. Juge said it definitely would not be marketed.

What is needed for general consumption in a market leaning more and more to the non-technical user, is not a more powerful language, but a simpler one.

"As far as a simpler interface for the user is concerned," he said, "a lot of people are working on that. If somebody comes up with it, I hope they will come to us first. We have four or five people working on it."

One important consideration of any new programming language is compatibility with what is already in the field.

Tandy finds making changes in hardware, for example, increasingly complex because of the compatibility problem.

When the Model III was designed, Juge said, "We gave up things like an 80 by 24 screen. . . to maintain compatibility with the Model I." As a result however, 80 percent of the programs for the Model I will run on the Model III without any changes, and most of the rest need only a minor adjustment, he said.

"Yet we've gotten just gobs of mail from people who are upset," Juge said.

Naw Modal I DOS

While he would give no details on it, Juge said, "A new Model I DOS is a distinct possibility."

However, he said, "If we come out with one we want to be sure it is compatible with what everyone is using."

According to Juge, if it isn't literally compatible with every single program, Radio Shack will junk it, even if they have invested a great deal of money in it.

Once a program like a new DOS is written, the problem becomes protecting that investment from copying.

Juge said Tandy is considering copy protection devices and legal action to protect what they regard as their property. Tandy put a simple device on its Model III programs to prevent copying, but they can be easily circumvented. Tandy promises a more effective system soon.

The new changes in the copyright laws which make it possible to copyright programs have not been tested in court, and, Juge said, until they are no one knows how effective they will be.

However, he said, at least in the case of TRSDOS Tandy is ready to go as far as necessary to protect its property.

TRSDOS is of particular concern because Tandy apparently suspects some independent software houses are putting it on disks with non-Radio Shack programs and without permission.

"We have objected to others using TRSDOS," he said. "We will take whatever legal means we have to protect it." He did not comment, however, on whether Tandy is actively preparing to sue over TRSDOS.

Juge labeled all of the TRS-80s basically business machines. Their penetration into the home market has been a sideline.

"We never built a home computer until the Color Computer," he said.

Actually, when Tandy entered the microcomputer market they did not know which sector of the market they would concentrate on.

"We didn't design the Model I as a business computer," Juge said. "We didn't know what we were building. We decided, 'That's dandy, let's build 3,500 of them and see if they sell.' "

If they didn't he said, Radio Shack would have put them in their own stores to use for inventory control.

In any event, when Tandy brought out the Model I for the first time at a convention, they all sold in a day. It was after that that Tandy began trying to find out who was using them.

"We have analyzed our user base pretty carefully," Juge said. "We never had the real sweatshirt, tennis shoe computer hobbyist, because he wants to play with the hardware, and we don't have any hardware that's really accessible."

About the threat of Japanese microcomputer competition, Juge is unconcerned.

"The Japanese companies that are making the most noise at present aren't the ones who have the distribution capabilities in the U.S.," he said. He referred to such companies as Sony or Panasonic as true competitors, if and when they enter the market.

Without distribution, he said, companies won't be able to get their products, however good, to the consumer. He did not comment directly into which category Sharp Electronics fell. Sharp, makers of the Pocket Computer, introduced its first microcomputer at the National Computer Convention in Chicago last month.■

by Bert Latamore
80 Staff

Color Computer Programs Promised

Color Computer owners, take heart. The software is coming.

Nineteen new programs for all areas of home use are being developed by Tandy/Radio Shack, Fort Worth, TX, according to Bill Walters, consumer information manager.

Ed Juge, Tandy's director of computer merchandising, said most of them will be brought out in the form of ROM (Read Only Memory) packs rather than on tape.

Walters said the programs will cover everything from keeping your personal accounts to games.

Tandy's policy is to produce only home use programs for the Color Computer, which they regard as strictly a home machine, Juge said.

The programs are in various stages of development, they said, and would be released individually as they are ready.

They declined to give details of any specific progams because some may not be released if certain bugs in them cannot be eliminated.■

Most people just sell disks. I sell you a complete system, and then I help you make it work.

It's called support, and it's a rare commodity in the microcomputer world.

It's also one big reason why they call my programs "the standard of the industry."

I'm Irwin Taranto, the one who changed the TRS-80* into a serious business computer. When you buy my TRS-80 systems (or, for that matter, one of my own computers that says "Taranto" on it), you buy me.

You buy my experience in making TRS-80 systems work in thousands of businesses around the world.

You buy the corrections, modifications and upgrades I constantly make on my TRS-80 systems.

And you buy my telephone number. You see, most of those thousand businesses needed a little help getting their systems up and going, and they called. We answered all their questions, and talked them through their problems. Every time the questions got really tough or really unusual, I'd answer them myself, on the phone, right then and there. I still do.

That pays off in two ways. It makes sure you get your systems working. It also alerts me to any little operating inefficiencies I might have designed into my systems. If there are any general business programs anywhere in the world, of any kind, that are checked out any better than my TRS-80 systems, I'd like to know about them.

I turned the TRS-80 into a serious computer.

The Model I, II and III business systems.

So far, I have six systems for the Model I, at $99 each:

Accounts Payable	General Ledger
Accounts Receivable	Payroll
Invoicing	Inventory Control

For the Cash Journal option on the General Ledger, add $50.

I also have six systems for the Model II:

General Ledger/Cash Journal	$ 299
Accounts Payable/Purchase Order	349
Open Accounts Receivable/Invoicing	349
Additional for Sales Analysis	100
Balance Forward Accounts Receivable	399
Payroll, without Job Costing	299
Additional for Job Costing	100
Inventory Control	399

For the Model III, we offer expanded versions of the six Model I systems, at $199 each.

Just call the number below and I'll send you any or all of the Model I or Model III systems by return mail. If you call about the Model II, I send you a questionnaire before I'll send you any systems. That lets me individualize the programs to your specific applications.

Why I call them "systems," not "programs."

There's a one-word answer: interaction. Each of the three sets of programs links to the General Ledger, and wherever it's useful, they cross-link to each other. For instance, "Sales Analysis" figures in a salesman's commission rate, so it links to "Payroll." Since it computes profitability within product categories, it links to "Invoicing."

That's what a system is. And that's one big difference between the Taranto TRS-80 business systems and somebody else's collection of business program disks.

If you like, I'll sell you the hardware, too.

I offer the TRS-80, Model II, along with selected peripherals. If you buy the computer from me, you get some extra advantages—hardware that's absolutely tailored to the programs, plus even more hand-holding from Taranto & Associates. The equipment won't cost you any more.

I can sell you a truly serious, completely supported, thoroughly proven business computer system for as little as $8000, hardware and software both.

There's nothing else like it in the market. Believe me, it's a far cry from that collection of program disks they're selling down the street.

✓45 **Taranto**
& ASSOCIATES, INC.

The Total System Store.™

121 Paul Drive, San Rafael CA 94903.
Outside California, toll free (800) 227-2868. In California, (415) 472-2670.
Authorized dealers throughout America.

*A trademark of the Tandy Corporation

NEW PRODUCTS

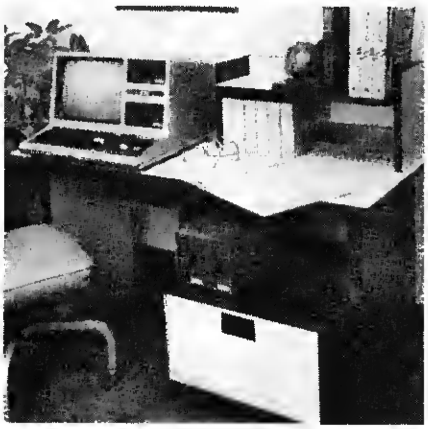
edited by Bert Latamore

Percom Sells Computer Desks

Percom Data Co. is selling custom designed, low cost computer furniture including a system desk with modular add-in and add-on components and a style-compatible printer stand.

The furniture, in TRS-80 compatible colors of pumice with black trim, is available from Percom Data Co., 211 N. Kirby, Garland, TX 75042.

Reader Service ✓ 336



Percom System Desk

New Language Out

PL/I-80 is a new all-purpose application programming language for 8080, Z80 and 8085 microprocessors.

The program has four major components: the PL/I-80 compiler and run time library; the LINK-80 linkage editor; and the RMAC relocatable macro assembler. It generates industry standard Microsoft relocatable codes so users can link load subroutines created by other language translators.

The package comes with three comprehensive manuals and a programmer's quick reference guide for $500 from Westico Inc., 25 Van Zant St., Norwalk, CT 06855.

Reader Service ✓ 347

Game Features Complex Maze

"Asmodeus" is a maze game for the TRS-80 Level II with 16K memory which features a world of 30 levels and more than 250 rooms which you explore in real time. You seek more than 500 treasures and deal with more than 500 monsters including Asmodeus, the ruler of evil.

The battle system includes six command options for attack or defense. It costs $39.95 from Serpent Software, 19 French Cres., Regina, Sask., S4R 6N3.

Readers Service ✓ 345

Three Challenges From Epyx

"The Datestones of Ryn", "Morloc's Tower" and "Rescue at Rigel" are available in an introductory three-pack of games from Epyx by Automated Simulations, Inc.

In "The Datestones of Ryn" the player has 20 minutes to find the Datestones and defeat the monsters of an underground maze where they have been hidden using 14 command options from firing an arrow to searching for secret doors.

In "Morloc's Tower" the player must catch Morloc in his maze before the wizard destroys the city.

In "Rescue at Rigel" the player has 60 minutes to find the 10 humans held captive somewhere inside an alien moonbase while dealing with monsters and armed robots and aliens.

The games are compatible with the 32K TRS-80. The kit costs $49.95 from Automated Simulations Inc., PO Box 4247 Mountain View, CA 94040.

Reader Service ✓ 338

Program Allows BASIC Editing

The Full Screen BASIC Text Editor from Computer Applications Unlimited allows insertions, deletions, copying and moving characters of lines or blocks of lines in Level II or disk BASIC on the Model I and Model III TRS-80.

The cursor can be moved to any position on the screen which can be scrolled up or down through the program text. Also included are global search and change commands and 26 user-definable "macro" keys.

The program is available for the Model I at $24.95 and the Model III at $29.95 from Computer Applications Unlimited, PO Box 214, Rye, NY 10580.

Reader Service ✓ 349

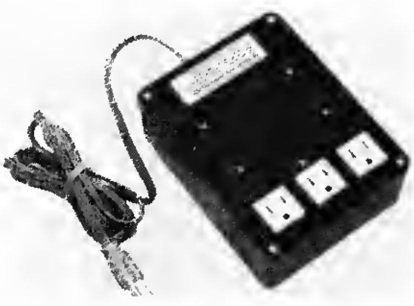
Power Pollution Controlled

The Super Isolator includes a heavy duty spike/surge suppressor and three individually dual-Pi filtered AC sockets to control electronic pollution for microprocessors.

The Super Isolator will control pollution for an 1,875 watt load with each socket handling a 1,000 watt load.

It is available from Electronic Specialists, Inc., 171 S. Main St., Natick, MA 01760 for $95.

Reader Service ✓ 328



Super Isolator

Disk Double Zap Available

The DOUBLE-ZAP/II program from Software, Etc., zaps NEWDOS 2.1/+ and

NEWDOS/80 for double density operation with the Percom Doubler.

It yields an extra 64,000 bytes on a diskette, will work with a one or two drive diskette in single and double density and will ready any single density TRSDOS or NEWDOS files and convert them to double density.

It is available for $39.95 for the NEWDOS 2.1/+and $49.95 for NEWDOS/80 from Software, Etc., 1839 Chamberlin Dr., Carrollton, TX 75007.

Reader Service ✓ 330

Quiet Dothead Lives Longer

The Micro-Nine-A printhead for Epson America, Inc.'s, MX-80 dot matrix printer reduces noise levels for the printer to below 60 dB with the addition of a sound damper around the unit and has a life expectancy of 100-million characters.

It retails for less than $40 from Epson America, Inc., 23844 Hawthorne Blvd., Torrance, CA 90505.

Reader Service ✓ 326

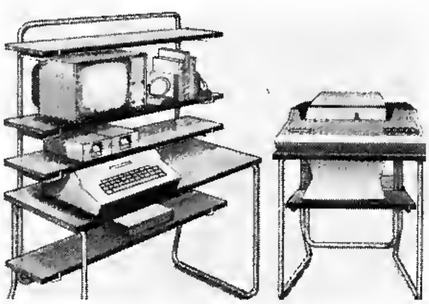
Control Program Primer Offered

CP/M Primer explains the control program for microcomputer operating systems for users of all levels of experience.

The book includes the latest CP/M version 2.0, a tear-out reference card, disk allocation and extents and a list of CP/M software.

The cost is $11.95 from Howard W. Sams & Co., Inc., 4300 W. 62nd St., Indianapolis, IN 46268.

Reader Service ✓ 329



Comstand Work Station

Computer Stand Marketed

The Comstand Computer Work Station for personal and small business computers creates a compact computer work area with an optimum table height with shelves above and below to hold monitors, disk drives and other hardware and reference books.

The Model 2036 is 36 inches long by 54 inches tall and the Model 2048 is 48 inches long by 47 inches tall. Both have 20 inch wide tables. A printer stand is also available.

The Model 2036 retails for $150, the Model 2048 for $165 and the printer stand for $140 from Ever Roll Specialties Co., 3988 Troy Road, Springfield, OH 45504.

Reader Service ✓ 339

Software Drives Disks

New OASIS software eliminates the need for the user-written disk driver by supporting the Cameo disk controllers which interface with a variety of 2.5- to 10-Mbyte cartridge disk drives, including the TRS-80 Model II, Tandy II and S-100.

Supplied with Cameo drivers at a nominal charge, OASIS is available separately for $500 for single-user and $850 for multiuser versions from Phase One Systems, 7700 Edgewater Dr., Suite 830, Oakland, CA 94621.

Reader Service ✓ 334

INV Controls Inventory

INV is an advanced inventory control system that keeps track of more than 2,000 items per disk on a 32K TRS-80 system.

The program is designed to produce minimum inventory investment, maximum customer service and low-cost plant operation at the same time.

INV is available from Micro Architect Inc., 96 Dothan St., Arlington, MA 02174 for $89.

Reader Service ✓ 333

Five Printers Introduced

Okidata Corporation has added a total of five new printers to its Microline and Slimline Series including a new high speed serial matrix printer, the Model 2350.

This machine prints bi-directionally at speeds to 350cps, features two-color printing and a program-controlled font selection. It does sub and superscripts, underscores and handles six standard, condensed and extra wide fonts.

The Microline 84 is the new top of the line model of Okidata's low cost Microline Series printers. It prints bi-directionally at 200 cps using a nine pin head for both text and graphics and can reproduce anything that can be displayed on a CRT screen.

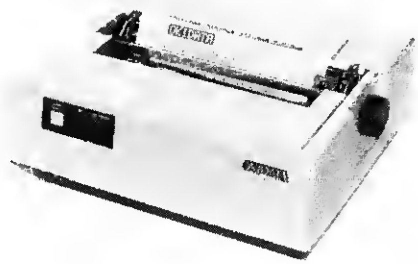
The Slimline MacroGrafix responds to simple commands to print tags, tickets, forms and lables.

The Microline 82A operates bi-directionally at 120 cps using a new nine pin head.

The Microline 83A is a wider printer, producing 136 columns on four-part forms to 15 inches in width at 120 cps.

Information on the printers is available from Okidata Corporation, 111 Gaither Drive, Mt. Laurel, NJ 08054.

Reader Service ✓ 337



Microline 82A Printer

TRS-80 with Typewriter

The KGS-80 keyboard actuator interfaces a TRS-80 with an IBM Selectric or SCM typewriter without modifying the typewriter.

The KGS-80 does not need software to operate and is fully compatible with popular word processing systems.

Information is available from NIK International Trading Inc., 114 Liberty St., Suite 204, New York, NY 10006.

Reader Service ✓ 343

Speak, Computer

You may not be able to talk to the animals, but with Cheaptalk your TRS-80 will talk to you.

Cheaptalk connects to an amplifier through the cassette output plug. Spoken words are digitized and stored in memory as self-contained subroutines, each re-

quiring 512 bytes of storage.

Cheaptalk includes a talking hex memory dump demonstration program and is available at $19.95 from Alan Saville, P.O. Box 5190, San Diego, CA 92105.

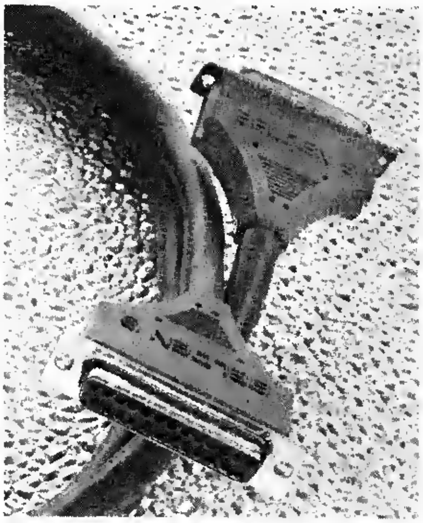
Reader Service ✓ 341

Review Magazine Started

The bi-monthly *Software Critic* reviews programs for the TRS-80 Models I and III for both experts and beginners without advertising.

The magazine is available at $15 per year from *Software Critic*, Box 3CH, University Park, NM 88003.

Reader Service ✓ 325



Shielded RS-232C Interface Cable Assemblies

Interface Cable Assemblies Introduced

Belden Corporation's new line of shielded interface cable assemblies comply with EIA Standard RS-232C and feature a 25-conductor cable in lengths from 5 to 70 feet and male or female subminiature D connectors mounted within a special molded handle.

The cable is designed to reduce bit error rate and maximize system performance and protect nearby equipment from signal radiation.

They are available from Belden Corp., Interconnect Systems Operation, 105 Wolfpack Rd., Gastonia, NC 28052.

Reader Service ✓ 184

Mince Edits Texts

Mince is a new video text editor for the TRS-80 Model II and other microcomputers based on the "Emacs" editor available on large minicomputer systems.

Mince allows you to move texts between files and manipulate it by word, sentence or paragraph as well as character, line and entire screen. It is compatible with 8080/Z80 micros, 16-bit micros and minicomputers allowing the user to upgrade his hardware while keeping the same text editor.

Mince is available for $125 for both software and documentation from Westico, Inc., 25 Van Zant St., Norwalk, CT 06855.

Reader Service ✓ 332

Program Tracks Grades

CLASSFILE is a classroom record keeping program for the 16K TRS-80 Model I or III that keeps track of up to 25 grades for up to 35 students.

It will: List all students in a class and their grades; list all students whose grades fall below a set cut-off point; list all students and their averages in rank order to the class average; and allow elimination of particular students from the class average. It also allows elimination of all grades to start a new term without retyping the names, and its companion, the CLASS ROSTER GENERATOR PROGRAM, formats the student list and prints a grading roster.

The program package is available for $19.95 from Teach Yourself By Computer Software, 40 Stuyvesant Manor, Geneseo, NY 14454.

Reader Service ✓ 342

Commercial Bulletin Board Uses Host TRS-80 Microcomputer

UEMS is a commerical oriented bulletin board system developed by Harry Lee which uses a host TRS-80 computer and allows billing users for their time on system and monitored access to individual pieces of mail.

The program stores messages of up to 2K bytes and maintains only pointer tables in memory, leaving more space for messages.

It costs $150 for three diskettes one containing the program and two containing data, and the user documentation. For an additional $150 you can obtain the commented source code and technical manuals from Small Business Systems Group Inc., 6 Carlisle Road, Westford, MA 01886.

Reader Service ✓ 183

Pamphlet Outlines Disk Care

Mr. Floppy teaches operators and users of floppy disks how to properly store and handle them in "Care and Feeding of Floppy Disks."

The updated version of the pamphlet is available from Advance Access Group Inc., 2200 S. Main St., Lombard, IL 60148.

Reader Service ✓ 185

Care and Feeding of Floppy Disks

Kit Details Software

Inotec's *Business Software Kit* gives several pages of data on a variety of business software packages to help businessmen select the programs they need.

It is available from Inotec, Inc., PO Box 1587, Clemson, SC, 29631 for $39.95.

Reader Service ✓ 327

See List of Advertisers on page 306

NEW PRODUCTS

Primer Teaches Pascal

Sams' *Pascal Primer* teaches hobbyists, novices and computerists the basics of Pascal in enough detail to allow the reader to write powerful Pascal programs.

Written by David Fox and Mitchell Waite it is available for $16.95 from Howard W. Sams & CO., Inc., 4300 W. 62nd St., Indianapolis, IN 46268.

Reader Service ✓ 329

Primer in Second Edition

Sams' Books has released a second edition of its *Microcomputer Primer* to reflect the latest developments in microcomputer technology in the business and personal areas.

Authors Mitchell Waite and Michael Pardee introduce the microcomputer to the reader and cover the central processing unit, memories, I/O interfacing, programming, operating systems and numbering sytems.

The Micorcomputer Primer costs $11.95 from Howard W. Sams & Co., Inc., 4300 W. 62 St., Indianapolis, IN 46268.

Reader Service ✓ 329

Financial Planning Program Announced

MINIMODEL is a financial planning program for the TRS-80 Model II with CP/M-like operating system.

It handles cash flow projections, financial forecasting, venture analysis, long range planning, project planning, risk analysis, and other financial tasks.

It costs $495 from Westico, Inc., 25 Van Zant St. Norwalk, CT 06855. Documentation alone costs $50.

Reader ✓ Service 348

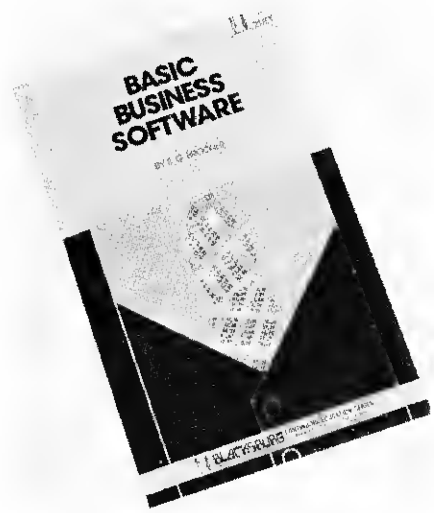
DOS Plus System Designed for Model III

DOS Plus is now usable on the Model III TRS-80 microcomputer. A double-density system, it is the first to be offered by a company outside of Radio Shack.

The disk operating system, which also works on the Model I, will run single or double-density on either machine.

It is available for $99 from Micro-Systems Software Inc., 5846 Funston St., Hollywood, FL. 33023

Reader Service ✓ 182



Basic Business Software

Book Teaches Businessmen

Basic Business Software by Ernest G. Brooner helps business people understand the basics of business software development with discussions of the three major subclasses of business programs, their uses and the languages they are written in.

Part of the Blacksburg Continuing Education Series, it is available for $9.95 plus $1 for shipping from Group Technology Ltd., PO Box 87, Check, VA 24072.

Reader Service ✓ 335

Filters Cut Noise

The Genisco C series dual function common mode filters eliminate both line-to-line and line-to-ground noise emissions to meet a wide range of international EMI specifications.

Information is available from Genisco Technology Corp., Components Division, 18435 Susana Road, Rancho Dominguez, CA 90221.

Reader Service ✓ 331

Cassette Quality Guaranteed

Micro-80 Computer Grade Cassettes are fully guaranteed 100 percent error free.

They are made of special formulation tape from industry leader Agfa-Gevaert of Germany encased in a five-screw construction shell of high impact styrene designed to maintain close tolerances for optimum tape to head alignment and control.

They are available from Micro-80 Inc., E-2665 North Busby Road, Oak Harbor, WA 98277.

Reader Service ✓ 350

Condominium Books Programmed

The Condominium Association Management (CAM) program is a general ledger accounting system for the TRS-80 Model II with a CP/M operating system, 48 K of memory, dual diskette drives, an 80 column by 24 line video terminal and an 80 column printer.

It will keep a running trial balance, balance sheet, profit and loss statement, budgeting statement, 12 month fee receipts history (by unit, name, address and telephone number), monthly fees, security deposits and additional fields that are user defined.

The checkwriter automatically prints vendor names and addresses on checks, allocates cash disbursements in up to nine different expense accounts and notifies the user when cash disbursements are greater than cash available. A customized accounts chart is also included.

A demonstration diskette with system manual is available for $35, which can later be applied to the full system price of $650, from A-T Enterprises, 221 N. Lois, La Habra, CA 90631.

Reader Service ✓ 344

Communications Program Links Micro, Time Share System

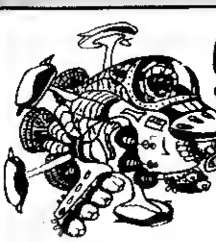
The ASCOM (ASyncronous COMmunication) communications program for TRS-80 Model II microcomputers with CP/M like operating systems can receive or transmit data files and can use various communication protocols, conversational and batch modes, system level commands for displaying directories and typing files to screen or printer.

The program and documentation costs $125 from Westico Inc., 25 Van Zant St., Norwalk, CT 06855.

Reader Service ✓ 346

See List of Advertisers on page 306

THE PROGRAM STORE



SPACE ACE 21

From Synergistic Solar
Build a better mousetrap -- or in this case spacecraft -- and see how well it works. In SPACE ACE 21, you design your own modular spaceship and see it graphically displayed and checked for errors. Then it's off into space to see how it fairs in battle against the computer's craft or that of another player. Choose from two or three dimensional galaxy and three different scenarios.

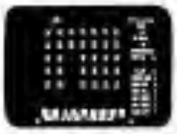
Also included: NEW STARSHIP VOYAGES
Battle the Rogues to defend a 3-D galaxy in this "Treck-type" space game.

16K TRS 80 I or III, 2-game pack...$19.95

CONFLICT 2500

From Avalon Hill
While checking your orbiting minefields, one of your Hyperfighters detects the presence of enemy Planet Pulverizers. Get set for a star battle in the 26th century. This new space game contains all the elements to hold your fascination: graphics, strategy, and challenge. Using the unique "Variable Scenario System" you set the size of your fleet and the level of complexity. Play one-on-one with the computer or with up to 10 players.

For 16K TRS-80, 16K Apple,
32K Atari...$14.95



INVADERS FROM SPACE

By Carl Miller from Acorn
A fast machine language approach to this classic (and addictive) space game. The aliens drop bombs and move from side to side trying to overrun your bases. You choose the speed, enemy bomb frequency and accuracy, your number of shots on screen and bases. Unlike most such games, you can move your base and simultaneously fire at the invaders. Full sound effects add even more excitement to the incredible action of INVADERS FROM SPACE. Fun for all ages and skill levels.

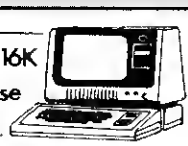
Protected tape...$14.95
Protected disk...$20.95

THE EMPIRE

From Computer Simulations
You have just captured the flagship of the evil Starforce Empire. Now you and your stalwart group must travel through the star system, ending the Empire's dictatorial rule. A combination of graphic display and tactical simulation lets you procure fighters and move throughout the galaxy.

16K tape...$14.95

TRS-80 Level II 16K unless otherwise noted



METEOR MISSION 2

By Hogue & Konyu from Big Five
Six astronauts are stranded on a desolate planet. You must undock from your command module and maneuver your rescue shuttle through the asteroid field to save them. You can only save one at a time, and each landing burns away parts of your landing sites. Order this realtime action game now or live with the astronauts' pitiful screams forever.

16K tape $14.95 (Mod I) $15.95 (Mod III)
For 32K disk....$17.95

SUPER NOVA

By Hogue & Konyu from Big Five
Asteroids surround your ship. You must shoot the asteroids, as well as any of the five types of alien spaceships. use your thrusters for full movement and rotation of your ship -- if you are overwhelmed, you can even jump to hyperspace! Written in fast machine code with superb graphics, this game is GREAT!

16K tape $14.95 (Mod I) $15.95 (Mod III)
For 32K disk....$17.95

DECISION MASTER
By David Brown from Syntonic
Decisions, decisions. Whether it's what time to get up, a change of jobs, buying that new house, or merging with Exxon, decisions have to be made, and DecisionMaster can help make the right ones. Using weighted factor analysis, Bayes' Rule, and present value theory, the program conducts a simple dialogue with you to determine a rational choice. Your first good decision comes right away -- when you decide to order DecisionMaster!

32K disk...$29.95

$$$ACQUIRE$$$

From Avalon Hill
A fantasy of laissez-faire capitalism. As an entrepreneur in the hotel business, your objective is to accumulate the greatest amount of capital wealth possible. You can play against the computer or with up to six players -- you can even let the computer play against itself! Buy and sell stock, arrange mergers and other maneuvers necessary to take you to the top! In ACQUIRE, the rich get richer... or do they?

For 48K TRS-80, 32K Apple...$19.95

PRE-SCHOOL I.Q. BUILDER

From PDI
A well executed series of programs, designed to aide in the development of prereading skills for three to five year olds. Starting with simple shapes, it develops the idea of "sameness" and "differentness" in form and spatial orientation. The shapes become progressively more complex, finally moving to "Letter Builder" where the child learns to differentiate the letters of the alphabet.

Tape...$14.95

SYSTEM SAVERS

By Tom Stibolt from Acorn
If you ever type "SYSTEM" on your TRS-80, this two-program package will make life easier for you.

FLEXL lets you make backup copies of any system format tape. This usually means easier loading than with machine-duplicated original tapes, and lets you store the original for security. Copies made using FLEXL display the filename of each program as it loads, for easier file searches.

Disk drive owners can use TDISK to save any system format tape onto disk. Many "system" programs cannot normally be loaded to disk, but with TDISK, they can. Why wait for slow tape loading? TDISK files load in seconds.

Two Program Pack...$14.95

OMNIKEY

From Discovery Bay
Use your energy for programming, not typing! Single keystroke entry of 26 Basic statements, auto-repeat on all keys, and selectable upper/lower case (if your keyboard is so equipped) speed Basic program entry immensely.

OMNIKEY's arrow-controlled cursor gives you full-screen editing capabilities. Insert or delete text at will. You can break multiple statements into separate program lines or combine them to save memory. Even repositioning of lines within a program is as easy as typing a new line number! Once you've used OMNIKEY, you won't program without it.

Tape (transferrable to disk)...$22.95

Visit Our Baltimore, MD Store: W. Bell Plaza – 6600 Security Blvd

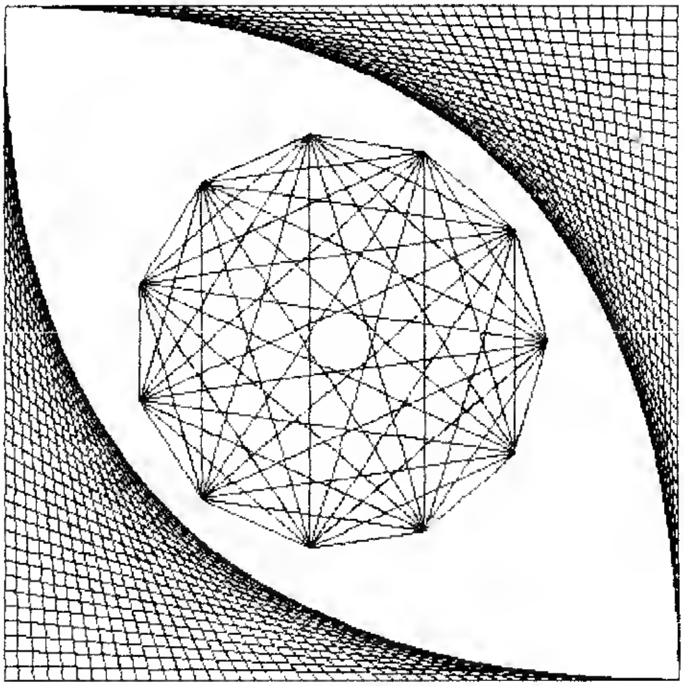
TO ORDER CALL TOLL FREE 800 424-2738 For information Call (202) 363-9797

THE PROGRAM STORE ✓17
4200 Wisconsin Avenue NW, Dept. K12 Box 9609
Washington, D.C. 20016

MAIL ORDERS: Send check or M.O. for total purchase price, plus $1.00 postage & handling. D.C. residents, add 6% tax. Charge card customers: include all embossed information on card.

Advanced Graphics Techniques—Part III

Bob Boothe
4651 Browndeer Lane
Rolling Hills Estate
CA 90274



Design created in two minutes, printed on a Base II dot matrix printer.

Part 2 of this series demonstrated how to create high resolution graphics with a dot matrix printer. Many readers probably could not use the programs because they don't own a printer with graphics capabilities; others may have converted the program to work on the video screen using low resolution graphics.

There is a better application for line drawing routines on the video. The program presented this month will manipulate representations of three dimensional objects on the video screen. The program also contains a routine which will print a design on the printer, and a routine which will reverse graphic characters on the screen.

All the views of an object this program produces have no hidden lines: All pictures will be optical illusions in which two different sides could be facing the viewer.

Program Listing 1 is a very simple BASIC routine which will draw a design similar to one of the designs from last month. The design is reproduced in Fig. 1. The routine at line 1500 can draw a line between any two points unless they form a vertical line. If you decide to use this BASIC routine instead of the machine language routine, the program can be easily modified to draw vertical lines.

The BASIC routine in Program Listing 1 took 36 seconds to draw Fig. 1 on the video screen. Dividing this time by 16, for 16 lines, I find it takes 2.25 seconds per line. Pretty fast right? Wrong. Now try the machine language BASIC combination. The same design using Program Listings 2 and 3 takes 2.4 seconds. Taking out line 1660 and running the program, I found that the BASIC portion of the program takes 1.9 seconds. The machine language portion took only a half a second for the entire design, or 0.031 seconds per line. That, my friends, is high-speed graphics.

Machine Language

This month, I will use four machine language routines, using four disk commands. First is the Line routine which draws a line between two points on the video screen. Next is the Save routine which produces hard copy on a Base 2 printer. The Field routine clears the screen with graphic blanks (128's) instead of normal blanks (32's). Finally, the Reset routine will reverse all the graphics on the screen. As a matter of fact, the Reset routine doesn't care what is on the screen: If the screen doesn't contain graphics, the program will produce a nice mess suitable for use as a secret code.

The Line routine is similar to last months program. The first section of the program automatically enables the disk commands by loading the new addresses directly into reserved RAM. In lines 220–270 two bytes are reserved for each variable. (Only two bytes are reserved because the largest number which can be used is 127, which fits comfortably into one byte with a bit to spare).

After the registers are saved in line 280,

"The program presented this month will manipulate representations of three dimensional objects on the video screen."

lines 290–370 compute and store the direction for the X coordinate. This process is repeated for the Y direction in lines 380–460. Lines 470–670 optimize the directions for maximum speed.

Lines 680–890 step across the line setting each block until the other end is met.

The part that differs is the Set routine. Last month, I had complete control over the organization of the individual points in each byte. This month the Set routine conforms to video memory. The easiest way to write a Set routine is not to write a routine at all, but to call the routine from ROM. However, since I have the old ROM and new computer users have the new ROM, I have written my own routine to set a point.

Whenever my routine is called, it first finds the point which needs to be set and puts it in the A and B registers. Lines 900 and 910 take care of this step. Lines 920–940 test to see if the X coordinate is odd or even. If it is odd, the E register is set to one. The E register will be used later to help determine which bit to set. Line 950 divides the X coordinate by two, because each byte is two blocks wide. This portion of the address is then put into the HL register pair.

Line 980 puts the Y coordinate into the accumulator. Lines 1000–1070 divide the Y coordinate by three and add 64 to the address in HL for each division. The remainder from the division is doubled and combined with the contents of the E register to form the bit which will be set. The bit pattern is then found in the table and combined with the display.

The Field routine is very simple. It puts a graphic blank in the first location of video memory and copies the blank through the entire block. A graphic blank is equal to 128.

```
5 REM  PROGRAM NO. 1  DEMONSTRATION OF ALL BASIC ROUTIN
     E
10 CLS
20 FORQ=1TO127STEP8
30 X1=Q:Y1=0
40 X2=0:Y2=47-Q/128*47
50 GOSUB 1500
60 NEXT
999 GOTO999
1500 REM  BASIC LINE ROUTINE FOR VIDEO
1600 FORA1=X1TOX2STEP(X2-X1)/SQR((X2-X1)*(X2-X1)+(Y2-Y1
     )*(Y2-Y1))
1610 SET(A1,Y1+(Y2-Y1)*(A1-X1)/(X2-X1))
1620 NEXT
1630 RETURN
```

Program Listing 1

This routine must be used instead of CLS because the normal clear screen routine uses blanks with a code of 32. If you want to see the problem, replace the Field command in the program with a CLS.

The Reset routine is almost as easy as the Field routine. The program takes each byte in video memory and executes an XOR 3FH. 3FH is equal to 0011 1111 in binary notation. The following is a demonstration of how this statement would affect a graphic block with a code of 93H:

93H = 1001 0011
3FH = 0011 1111

ACH = 1010 1100

The XOR statement tests each pair of bits; if they are different, the result is one. If the two bits are the same, the result is zero. You can verify that 93H and ACH are opposite graphic characters by finding them on a chart (93H = 147, ACH = 172).

According to my precise measurements, the screen is 7.5 inches wide by 6.75 inches high. The screen is 128 graphic blocks wide by 48 graphic blocks high. By feeding all these numbers to my number cruncher, I found that graphic blocks are exactly 2.4 times higher than they are wide. In simpler terms, they are pretty skinny. The problem is to make a block on the printer that is the same shape as one on the screen.

The Base 2 printer can print in six different character densities: 64, 72, 80, 96, 120 and 132 characters per eight inch line. My first intention was to make the blocks three by four dots, as they are on the screen. The printer prints 72 dots per inch vertically, and the block will be four dots. This means that the blocks should be .0556 inches long by .0232 inches wide. If the blocks are three dots wide, the 132 characters per line mode makes each block .030 inches wide. This is too wide, but it is as close as possible.

When I wrote the program accordingly, I discovered a problem: The printer can't handle high density graphics. Many of the lines would fade and even disappear. The manual

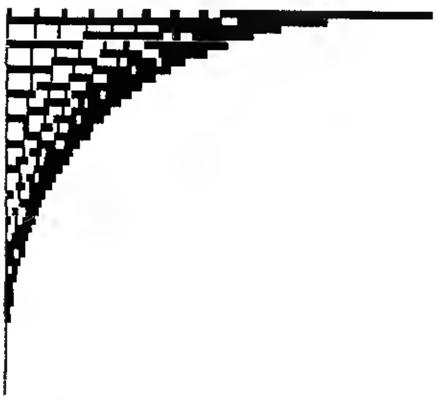


Fig. 1. This pattern can be created with Program 1 or 3.

```
  10 REM  PROGRAM NO. 2    CALLING ROUTINE FOR MACHINE LA
       NGUAGE
  20 FIELD
  30 PI=3.1416
1000 GOTO1000
1500 REM LINE ROUTINE
1520 IF X1 < 0 OR X1 > 127 THEN PRINT"X1 ILLEGAL":END
1530 IF X2 < 0 OR X2 > 127 THEN PRINT"X2 ILLEGAL":END
1540 IF Y1 < 0 OR Y1 > 47 THEN PRINT"Y1 ILLEGAL":END
1550 IF Y2 < 0 OR Y2 > 47 THEN PRINT"Y2 ILLEGAL":END
1560 IF INT(X1)=INT(X2) AND INT(Y1)=INT(Y2) THEN SET(X1
     ,Y1):RETURN
1600 POKE32331,X1
1610 POKE32333,Y1
1620 POKE32335,X2
1630 POKE32337,Y2
1660 LINE
1680 RETURN
```

Program Listing 2

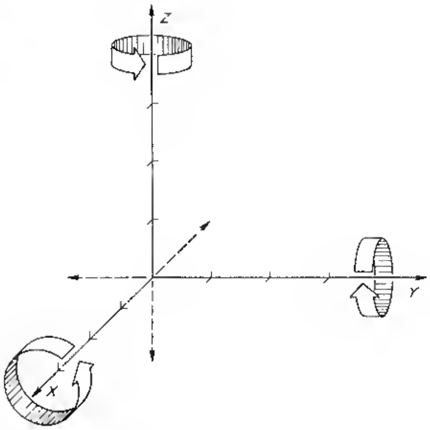


Fig. 2. This shows the direction of each axis, and the direction of rotation about each axis.

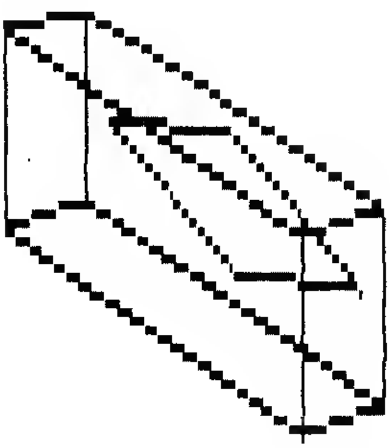


Fig. 3. This is a view of a tissue box on its side.

said that the printer was getting too hot. I said, "How can it get too hot on the first line?" The darn manual didn't even reply.

I decided I would need to add a time delay to let the printer cool off between lines. The original screen print time was about two and a half minutes long. My first time delay extended the total time to five minutes. Another later delay brought the total time up to seventeen minutes. Did it work? Of course not. Actually, I did get one good printout with the seventeen minute routine (Fig. 1).

My next routine was for the 64 CPL mode which is 48 DPI. This means that each block will be 2 × 7 dots. I took out most of the delay because I didn't want to wait more than five minutes for a picture. Fortunately, this routine has worked every single time.

The routine starts as all the other routines, by exchanging the registers. Lines 1520–1610 select 64 CPL and set the vertical spacing at seven dots per line. Line 1620 sets the HL register pair to the start of video memory, and line 1630 initializes the loop for 16 lines of printing.

The output loop is more difficult because each byte covers parts of three different printed lines. My printing section has three subsections called TOPLIN, MIDLIN and BOTLIN which print the top, middle and bottom lines respectively.

Each subsection first calls TRSMIT, which outputs the control sequence to transmit graphics and some leading blanks, and then calls Delay, which gives a delay of a few seconds.

The TOPLIN routine in line 1690 tests bit 0, which is the top left bit, and then calls Spot. Spot prints either a blank or a graphic block. Bit 1 is tested next; it is the top right bit. The spot routine is called again, and this process is repeated for all 64 bytes on the line. Finally, Blanks is called, which fills the remainder of the line with blanks and outputs a carriage return to cause a line feed.

After the top line is finished, HL is set back to the start of the line, and the entire procedure is repeated for bits two and three. BOTLIN repeats this one more time for bits four and five. After BOTLIN is finished, the outside loop counter is decremented, and the loop is continued until all 16 lines from the video have been printed.

Each line in the 64 CPL mode requires 384 bytes of graphical data. The display needs only 256 bytes, and 128 blanks have to be sent. I decided to send 64 before the data and 64 after it, which centers the printout on the paper. The TRSMIT routine outputs the control sequence to inform the printer to accept graphical data, and it sends the first 64 blanks. The Blanks routine outputs the other 64 blanks and sends a carriage return. The Blanks routine uses 128 as a blank instead of zero because the printer does not recognize zeros.

PRTOUT is a routine from the T-BUG manual. The Spot routine outputs either a solid block if the Z flag is not set, or a blank if the Z flag is set. The Delay routine can be modified to waste more time by changing the number of loops in line 2340. If you want to create hard copy with a printer other than the Base 2, you may need to make some modifications to the routine. If you do change the program, be sure the last address does not exceed 7FFFH.

Memory Size 32330

Create and load the machine language portion of the program first. Do not forget to set memory size to 32330.

Listing 2 is the routine which calls the machine language Line program. To use this program, you should write your BASIC program between lines 30 and 1000. Listing 3 is a very simple example of the kind of routine which can be used with this program. It draws the same shape as Listing 1. Any of the programs from last month's article can also be used if they are modified to fit on a 128 × 48 dot screen. To write your own design program, you should compute the endpoints of each line using the variables X1, Y1, X2 and Y2. After the values are assigned, a GOSUB 1500 will draw the line.

Listing 2 checks the values of the points to see if they are in the range, and if the two points are the same. If the points are out of range, an error message is displayed. If the points are the same, one point is set, and the program is continued. If everything is satisfactory, the points are POKEd into the

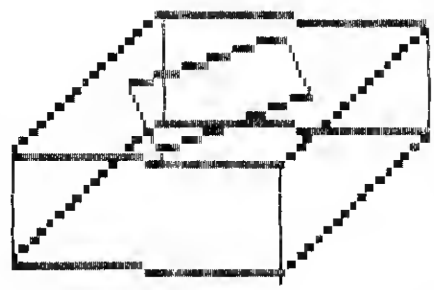


Fig. 4. This is a view looking down the box from the end.

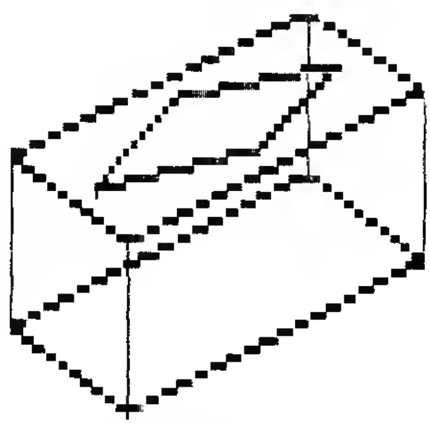


Fig. 5. This is the economy size pack, stretched three times in the Z direction.

SOFTWARE UNLIMITED

presenting the LARGEST SELECTION OF SOFTWARE EVER ASSEMBLED...

the best available on the market today for TRS-80® Computers

at SUPER DISCOUNT PRICES!

AVALON HILL

- ☐ MIDWAY 13.50
- ☐ NUKE WAR 13.50
- ☐ PLANET MINERS 13.50
- ☐ CONVOY RAIDER 13.50
- ☐ B1 BOMBER 13.50
- ☐ LORDS OF KARMA 18.00
- ☐ CONFLICT 2500 .... 13.50
- ☐ COMPUTER ACQUIRE 18.00

APPARAT

- ☐ NEWDOS/80 .... 125.00

ADVENTURE INTERNATIONAL

- ☐ ADVENTURE HINT BOOK 7.95
- ☐ ADVENTURE #0 .... 6.25
- ☐ ADVENTURE (1,2,3) [D] 35.95
- ☐ ADVENTURE (4,5,6) [D] 35.95
- ☐ ADVENTURE (7,8,9) [D] 35.95
- ☐ ADVENTURE #10 [D] 18.95
- ☐ ADVENTURE (specify 1-10) 13.55
- ☐ MEAN CHECKERS MACHINE 17.95
- ☐ MEAN CHECKERS MACHINE [D] 22.95
- ☐ DR. CHIPS 13.55
- ☐ DR. CHIPS [D] 17.95
- ☐ INTER FICTION SAMPLER [D] 13.55
- ☐ INTER-LOCAL CALLS [D] 17.95
- ☐ INTER-TWO HEADS [D] 17.95
- ☐ INTER-IMPETUOUS [D] 17.95
- ☐ KID-VENTURE 1 .... 17.95
- ☐ STAR TREK 3.5 .... 13.55
- ☐ STAR TREK 3.5 [D] 17.95
- ☐ ZOSSED IN SPACE 13.55
- ☐ MACES #1, BARLOG 31.00
- ☐ MACES #1, SISYPHUS 31.00
- ☐ LUNAR LANDER 13.55
- ☐ LUNAR LANDER [D] 18.95
- ☐ POKER 13.55
- ☐ GALACTIC TRILOGY [D] 35.95
- ☐ SILVER FLASH 13.55
- ☐ SILVER FLASH [D] 17.95
- ☐ MISSILE ATTACK 13.55
- ☐ STAR SCOUT 13.55
- ☐ SPACE COLONY [C] 8.95

ACORN SOFTWARE

- ☐ ATERM 17.95
- ☐ SYSTEM SAVERS 13.55
- ☐ DISASSEMBLER 13.55
- ☐ DISK/TAPE UTILITY 17.95
- ☐ STAR TREK SIMULATION 8.95
- ☐ GAMMON CHALLENGER 13.55
- ☐ PIGSKIN 13.55
- ☐ ULTRA TREK 13.55
- ☐ SPACE WAR 8.95
- ☐ WARP/LANDER 8.95
- ☐ BASKETBALL [D] 18.95
- ☐ BASKETBALL 13.55
- ☐ DUEL-N-DROIDS [D] 18.95
- ☐ DUEL-N-DROIDS 13.55
- ☐ INVADERS FROM SPACE 13.55
- ☐ INVADERS FROM SPACE [D] 18.95
- ☐ PIGSKIN [D] 18.95
- ☐ PINBALL 13.55
- ☐ PINBALL [D] 18.95
- ☐ EVEREST EXPLORER 13.45
- ☐ EVEREST EXPLORER [D] 18.95
- ☐ SUPERSCRIPT [D] 28.95

LOBO

- ☐ ELDOS [D] 125.00

MED SYSTEMS

- ☐ DEATHMAZE 5000 .... 11.65
- ☐ RATS REVENGE 11.65
- ☐ REALITY ENDS 8.95

HAYDEN

- ☐ SARGON II 25.00
- ☐ SARGIN II [D] 30.00

AUTOMATED SIMULATION

- ☐ STAR WARRIOR [CASS. OR DISK] 35.95
- ☐ THREE PACK [D] 45.00
- ☐ STARFLEET [CASS. OR DISK] 22.50
- ☐ INVASION [CASS. OR DISK] 22.50
- ☐ APSHAI [CASS. OR DISK] 35.95
- ☐ RYN (CASS., OR DISK) 17.95
- ☐ MORLOC [CASS. OR DISK] 17.95
- ☐ RIGEL [CASS. OR DISK] 26.95
- ☐ HELLFIRE [CASS. OR DISK] 35.95

BIG FIVE SOFTWARE

- ☐ ATTACK FORCE (MODEL 1 OR 3) 14.30
- ☐ GALAXY INVASION (MODEL 1 OR 3) 14.30
- ☐ METEOR MISSION II (MODEL 1 OR 3) 14.30
- ☐ SUPER NOVA (MODEL 1 OR 3) 14.30
- ☐ COSMIC FIGHTER (MODEL 1 OR 3) 14.30

COMPUTER SIMULATIONS COMPANY

- ☐ BATTLE OF BULGE-BASTGONES 17.95
- ☐ D-DAY INVASION OF FRANCE 17.95
- ☐ DARK KINGDOM 11.75
- ☐ DOG RACE, COLOR ONLY [C] 5.95
- ☐ ELECTRIC ALARM 4.95
- ☐ EMPIRE STRIKES BACK 13.55
- ☐ GREAT DICTATOR 7.25
- ☐ JEDI KNIGHT 11.75
- ☐ LUNAR ENCOUNTER 11.75
- ☐ MERCENARY 8.95
- ☐ MICRO ARCADE 13.55
- ☐ SHARK, COLOR ONLY [C] 5.95
- ☐ SLOT MACHINE 7.25
- ☐ BATTLE OF BULGE-ST. VITH 13.55
- ☐ STAR CRUISERS 13.55
- ☐ TYPE WRITERS 17.95
- ☐ U-BOAT, COLOR ONLY [C] 5.95
- ☐ ORION WAR, COLOR ONLY [C] 5.95
- ☐ POLARIS, COLOR ONLY [C] 5.95
- ☐ ASTEROID PIRATES 13.55

INSTANT SOFTWARE

- ☐ AIR FLIGHT SIMULATION 7.25
- ☐ AIRMAIL PILOT 8.95
- ☐ ALL STARS [D] 31.55
- ☐ ASTEROIDS [D] 17.95
- ☐ ASTEROIDS 13.55
- ☐ BASIC PROGRAMMING ASSISTANT 13.55
- ☐ BALL TURRET GUNNER 8.95
- ☐ BATTLEGROUND 8.95
- ☐ CHECK MANAGEMENT [D] 36.55
- ☐ CHESSMATE-80 .... 17.95
- ☐ THE COMMUNICATOR 8.95
- ☐ COSMIC PATROL 13.55
- ☐ COSMIC PATROL [D] 17.95
- ☐ DAREDEVIL 8.95
- ☐ DISASSEMBLER 8.95
- ☐ DISK EDITOR [D] 36.55
- ☐ DISK SCOPE [D] 17.95
- ☐ DLDIS [D] 17.95
- ☐ DEVICE DRIVERS 17.95
- ☐ ENHANCED BASIC 22.50
- ☐ ELEC REPRT GEN 8.95
- ☐ FLIGHT PATH 8.95
- ☐ FLYING CIRCUS [D] 36.55
- ☐ GEOGRAPHY [D] 45.00

INSTANT SOFTWARE

- ☐ I.Q. TEST 8.95
- ☐ INVADERS 8.95
- ☐ INVESTORS PARADISE 8.95
- ☐ IRV 22.50
- ☐ IRV [D] 27.55
- ☐ JET FLIGHTER PILOT 13.55
- ☐ MIND WARP 8.95
- ☐ MONEY MADNESS 8.95
- ☐ NIGHT FLIGHT 8.95
- ☐ ONE-D MAILING LIST [D] 22.50
- ☐ OTHELLO 8.95
- ☐ PROGRAMMER'S CONVERTER 8.95
- ☐ QSL MANAGER [D] 17.95
- ☐ SANTA PARAVIA FIUMACCIO 8.95
- ☐ SKIRMISH-80 .... 8.95
- ☐ TERMINAL-80 .... 22.50
- ☐ TLDIS 13.55
- ☐ UTILITY I 7.25
- ☐ UTILITY II 7.25
- ☐ WINNERS DELIGHT 8.95
- ☐ WORDSLINGER 26.00

PERSONAL SOFTWARE

- ☐ CCA MGMT-TRS-80 .... 65.00
- ☐ MONTY SPACE MONOPOLY [D] 35.95
- ☐ ZORK [D] 35.95

QUALITY SOFTWARE

- ☐ DEBUG 13.55
- ☐ FASTGAMMON 17.95
- ☐ LOWBALL POKER 10.45
- ☐ BANKSHOT 8.95
- ☐ POKER PETE 10.45
- ☐ RUMMY MASTER 10.45
- ☐ 3D TIC TAC TOE 13.55
- ☐ QS LIGHT PEN 17.95
- ☐ SKETCH 80 .... 13.55

STRATEGIC SIMULATIONS

- ☐ COMPUTER BISMARCK [D] 51.50
- ☐ COMPUTER BISMARCK 42.00

SUB-LOGIC

- ☐ T80-FS1 FLIGHT SIMULATOR 22.00
- ☐ 3D GRAPHICS 26.50

BOTTOM SHELF

- ☐ ANALYSIS PAD [D] 90.00
- ☐ BASIC TOOLKIT 17.80
- ☐ BUSINESS MAIL [D] 100.00
- ☐ CHECKBOOK II [D] 44.50
- ☐ CHECK REGISTER [D] 67.00
- ☐ HEAD CLEANER [D] 17.00
- ☐ INFO SYSTEM [D] 44.50
- ☐ LIBRARY 100 .... 44.50
- ☐ SYSTEM DOCTOR [D] 35.50
- ☐ SYSTEM DOCTOR 26.00

MICROSOFT SOFTWARE

- ☐ ADVENTURE [D] 25.50
- ☐ ASSEMBLY DEVELOPMENT [D] 80.00
- ☐ BASIC COMPILER [D] 175.00
- ☐ EDITOR/ASSEMBLER 25.00
- ☐ FORTRAN COMPILER [D] 80.00
- ☐ LEVEL III BASIC 44.00
- ☐ MuMATH [D] 64.00
- ☐ OLYMPIC DECATHALON (CASS OR DISK) 20.00
- ☐ TYPING TUTOR 13.55

If you don't see it listed, write...we probably have it in stock!

Check program desired.
Complete ordering information
and mail entire ad.
Immediate Shipment from stock.

Ship the above programs as checked to:

Mr./Mrs. ______________________

Address ______________________

City ______________________

State ______________ Zip ______

I have a ______________________ (name of Computer)

with ______________ k memory

Number of Programs Ordered .... ________

Amount of order .... ________

N.Y. residents add Sales Tax .... ________

Add shipping anywhere in the U.S. 2.00

Total amount enclosed .... ________

Charge my: ☐ Master Charge ☐ Visa

Signature ______________________

Card No. ______________ Expires ______________

Personal Checks please allow 3 weeks.

KEY:
C-TRS-80 Color
D-on Disc
If not marked - Cassette

Mail to: **DIGIBYTE SYSTEMS CORP.**

31 East 31st Street, New York, N.Y. 10016
(212) 889-8975

TRS-80 is a registered trademark of TANDY CORP.

Prices subject to change without notice.

✓226

Program Listing 3

```
                  00100 ;LINE DRAWING ROUTINE FOR 16K LEVEL II ON VIDEO
                  00110 ;SCREEN PRINT ROUTINE IS FOR BASE II PRINTER
                  00120 ;BOB BOOTHE 8-12-80
41A3              00130          ORG      16804-1        ;LINE LOCATION
41A3 C3567E       00140          JP       LINE           ; AUTOMATICALLY ENABLE
41A0              00150          ORG      16801-1
41A0 C33D7F       00160          JP       SAVE
417C              00170          ORG      16765-1
417C C31A7F       00180          JP       FIELD
419A              00190          ORG      16795-1
419A C32A7F       00200          JP       RSET
7E4A              00210          ORG      32330          ;SO END IS 7FFFH
7E4A 0000         00220 STARTX   DEFW     0       ;RESERVE 2 BYTES FOR EACH VARIABLE
7E4C 0000         00230 STARTY   DEFW     0
7E4E 0000         00240 ENDX     DEFW     0
7E50 0000         00250 ENDY     DEFW     0
7E52 0000         00260 DIRX     DEFW     0
7E54 0000         00270 DIRY     DEFW     0
7E56 D9           00280 LINE     EXX                     ; SAVE REGISTERS
7E57 2A4E7E       00290          LD       HL,(ENDX)      ;GET END OF LINE
7E5A ED5B4A7E     00300          LD       DE,(STARTX)    ;GET START
7E5E 6C           00310          LD       L,H            ;PUT MSB'S INTO LSB'S
7E5F 2600         00320          LD       H,0            ;MAKE MSB'S 0
7E61 5A           00330          LD       E,D
7E62 1600         00340          LD       D,0
7E64 B7           00350          OR       A              ;RESET CARRY FLAG
7E65 ED52         00360          SBC      HL,DE          ;COMPUTE DIRECTION
7E67 22527E       00370          LD       (DIRX),HL      ;SAVE DIRECTION
7E6A 2A507E       00380          LD       HL,(ENDY)      ;REPEAT FOR Y DIRECTION
7E6D ED5B4C7E     00390          LD       DE,(STARTY)
7E71 6C           00400          LD       L,H
7E72 2600         00410          LD       H,0
7E74 5A           00420          LD       E,D
7E75 1600         00430          LD       D,0
7E77 B7           00440          OR       A
7E78 ED52         00450          SBC      HL,DE
7E7A 22547E       00460          LD       (DIRY),HL
7E7D 2A527E       00470          LD       HL,(DIRX)      ;OPTIMIZE DIRECTIONS
7E80 ED5B547E     00480          LD       DE,(DIRY)
7E84 7D           00490          LD       A,L            ;STORE BIT 7'S
7E85 E680         00500          AND      80H
7E87 47           00510          LD       B,A
7E88 7B           00520          LD       A,E
7E89 E680         00530          AND      80H
7E8B 4F           00540          LD       C,A
7E8C CB25         00550 SHIFT    SLA      L              ;SHIFT UNTIL 7'S CHANGE
7E8E CB23         00560          SLA      E
7E90 7D           00570          LD       A,L
7E91 E680         00580          AND      80H
7E93 B8           00590          CP       B
7E94 2008         00600          JR       NZ,DSHIFT      ;IF NOT SAME DONE SHIFT
7E96 7B           00610          LD       A,E
7E97 E680         00620          AND      80H
7E99 B9           00630          CP       C
7E9A 2002         00640          JR       NZ,DSHIFT
7E9C 18EE         00650          JR       SHIFT          ;SHIFT 'EM AGAIN
7E9E 22527E       00660 DSHIFT   LD       (DIRX),HL      ;SAVE THE NEW DIRECTIONS
7EA1 ED53547E     00670          LD       (DIRY),DE
7EA5 CDDA7E       00680 NXTBLO   CALL     SET            ;SET THE BLOCK
7EA8 AF           00690          XOR      A
7EA9 2A4E7E       00700          LD       HL,(ENDX)      ;DOES START = END  YET
7EAC ED5B4A7E     00710          LD       DE,(STARTX)
7EB0 ED52         00720          SBC      HL,DE
7EB2 2818         00730          JR       Z,MAYBE
7EB4 2A4A7E       00740 NOTYET   LD       HL,(STARTX)    ;ADD DIRECTION TO START
7EB7 ED5B527E     00750          LD       DE,(DIRX)
7EBB 19           00760          ADD      HL,DE
7EBC 224A7E       00770          LD       (STARTX),HL
7EBF 2A4C7E       00780          LD       HL,(STARTY)
7EC2 ED5B547E     00790          LD       DE,(DIRY)
7EC6 19           00800          ADD      HL,DE
7EC7 224C7E       00810          LD       (STARTY),HL
7ECA 18D9         00820          JR       NXTBLO
7ECC B7           00830 MAYBE    OR       A              ;RESET CARRY FLAG
7ECD 2A507E       00840          LD       HL,(ENDY)      ;LINE COULD BE VERTICAL
7ED0 ED5B4C7E     00850          LD       DE,(STARTY)
7ED4 ED52         00860          SBC      HL,DE
7ED6 20DC         00870          JR       NZ,NOTYET
```

Program continues

"Most readers will notice that the bolt has a hexagonal shaft rather than a round one; well, let's see you try to draw a circle with straight lines."

```
7ED8 D9                   EXX                        ;RETURN WITH OLD REG'S
7ED9 C9                   RET
7EDA 3A4B7E        SET    LD     A,(STARTX+1)        ;SET PIXEL (A,B)
7EDD ED4B4C7E             LD     BC,(STARTY)         ;GET A AND B FROM MSB'S
7EE1 1EØØ                 LD     E,Ø
7EE3 CB47                 BIT    Ø,A                 ;IS A ODD
7EE5 C4187F               CALL   NZ,ADDOFF           ;BIT WILL BE ODD
7EE8 CB3F                 SRL    A                   ;DIVIDE BY 2
7EEA 6F                   LD     L,A                 ;MAKE CHARACTER POSITION
7EEB 26ØØ                 LD     H,Ø                 ;IN HL = Ø TO 64
7EED 78                   LD     A,B                 ;WORK ON Y DIRECTION
7EEE D5                   PUSH   DE                  ;SAVE E
7EEF 114ØØØ               LD     DE,64               ;64 CHARACTERS PER LINE
7EF2 B7                   OR     A                   ;RESET CARRY FLAG
7EF3 ED52                 SBC    HL,DE               ;SUBTRACT ONE LINE
7EF5 C6Ø3                 ADD    A,3                 ;ADD ONE LINE SO A>3
7EF7 D6Ø3          BIT    SUB    3                   ;DIVIDE BY 3
7EF9 19                   ADD    HL,DE               ;ADD A LINE EACH TIME
7EFA FEØ3                 CP     3                   ;IS A<3 YET
7EFC 3ØF9                 JR     NC,BIT              ;IF NO, KEEP SUBTRACTING
7EFE D1                   POP    DE                  ;GET OLD PART OF BIT
7EFF CB27                 SLA    A                   ;DOUBLE REMAINDER
7FØ1 83                   ADD    A,E                 ;ADD OLD REMAINDER
7FØ2 11ØØ3C               LD     DE,1536Ø            ;START OF VIDEO
7FØ5 19                   ADD    HL,DE               ;COMPUTE ADDRESS
7FØ6 47                   LD     B,A                 ;PUT BYTE IN B
7FØ7 Ø4                   INC    B                   ;SO NOT Ø
7FØ8 11117F               LD     DE,TABLE-1
7FØB 13            FINDT  INC    DE                  ;FIND POSITION IN TABLE
7FØC 1ØFD                 DJNZ   FINDT
7FØE 1A                   LD     A,(DE)
7FØF B6                   OR     (HL)
7F1Ø 77                   LD     (HL),A
7F11 C9                   RET
7F12 Ø1            TABLE  DEFB   Ø1H                 ;BIT Ø
7F13 Ø2                   DEFB   Ø2H                 ;BIT 1
7F14 Ø4                   DEFB   Ø4H
7F15 Ø8                   DEFB   Ø8H
7F16 1Ø                   DEFB   1ØH
7F17 2Ø                   DEFB   2ØH
7F18 1C            ADDOFF INC    E                   ;LET E=1 FOR ODD BIT
7F19 C9                   RET
7F1A D9            FIELD  EXX                        ;SAVE REGISTERS
7F1B 21ØØ3C               LD     HL,1536Ø            ;START OF VIDEO
7F1E 11Ø13C               LD     DE,15361            ;START + 1
7F21 Ø1FFØ3               LD     BC,1Ø23             ;NUMBER OF BYTES
7F24 368Ø                 LD     (HL),128            ;GRAPHIC BLANK
7F26 EDBØ                 LDIR                       ;COPY
7F28 D9                   EXX
7F29 C9                   RET
7F2A D9            RSET   EXX                        ;REVERSE VIDEO ROUTINE
7F2B Ø1ØØØ4               LD     BC,1Ø24             ;NUMBER OF BYTES
7F2E 21ØØ3C               LD     HL,1536Ø            ;START OF VIDEO
7F31 7E            REVERS LD     A,(HL)
7F32 EE3F                 XOR    3FH                 ;ØØ11 1111 BINARY
7F34 77                   LD     (HL),A
7F35 23                   INC    HL
7F36 ØB                   DEC    BC                  ;DECREMENT COUNTER
7F37 78                   LD     A,B                 ;CHECK IF DONE
7F38 B1                   OR     C
7F39 2ØF6                 JR     NZ,REVERS           ;REPEAT TIL DONE
7F3B D9                   EXX
7F3C C9                   RET
7F3D D9            SAVE   EXX                        ;SCREEN PRINT ROUTINE
7F3E 3E1B                 LD     A,27                ;ESCAPE
7F4Ø CDD47F               CALL   PRTOUT
7F43 3E64                 LD     A,1ØØ               ;64 CHAR PER LINE
7F45 CDD47F               CALL   PRTOUT
7F48 3E1B                 LD     A,27                ;ESCAPE
7F4A CDD47F               CALL   PRTOUT
7F4D 3E62                 LD     A,98                ;SET VERTICAL LINE
7F4F CDD47F               CALL   PRTOUT
7F52 3EØE                 LD     A,14                ;SPACING TO 14 HALF DOTS
7F54 CDD47F               CALL   PRTOUT
7F57 21ØØ3C               LD     HL,1536Ø
7F5A Ø61Ø                 LD     B,16                ;NUMBER OF LINES
7F5C C5            LOOP   PUSH   BC
7F5D CDB37F               CALL   TRSMIT
7F6Ø CDEE7F               CALL   DELAY
7F63 114ØØØ               LD     DE,64               ;CHARACTERS PER LINE
7F66 Ø64Ø                 LD     B,64                ;64 CHARACTERS PER LINE
7F68 CB46          TOPLIN BIT    Ø,(HL)              ;TEST FOR BLOCK
```

Program continues

Your best Model III peripheral buy is a modem.

Your best Model III modem buy is LYNX.

OK, you've familiarized yourself with your new TRS-80 Model III, and you're ready to expand.

You've got plenty of peripherals to choose from.

Disk drives, voice boxes, printers, joy sticks—the list is almost endless. And all may help you get more enjoyment out of the info you put in.

But there's the catch. Your Model III still relies solely on you for input. Without you to write increasingly complex programs—or pay cold cash to buy them—it's blind, deaf and dumb.

The real expansion will begin only when you give it ready access to the larger world of data communications. The world of free-access "bulletin board" resources. The world of electronic mail, instant news and financial reports and games from vast, affordable services like Source and CompuServe. The world of thousands of computer people just like you, in homes and businesses around the block and across the country.

It's a world you can tap through your telephone . . . but only if you're properly equipped.

You need a modem. And not just any modem.

You need LYNX.

The new LYNX is the ideal modem for your Model III. That's because it's the key element of a total direct-connect telephone communications package.

This package—including serial and computer bus interfaces, cables and terminal software—is available to you for a remarkable **SINGLE PRICE: $299.95.**

Compared to other modems, LYNX is superior. It doesn't make you buy adapter cables or circuit boards separately to bring it on line. It provides a full one-year factory warranty, instead of the "normal" 90 days.

It includes performance features—like **auto dial/answer,** programmable I/O porting, variable word length, parity and stop bits—that are either optional at extra cost or unavailable with other modems. It requires no tools to install.

It can be used with any TRS-80 communications programs; it's not restricted to its own special software. A full range of easy-to-find ST-80 cassette and disk software authored by Lance Micklus is available nationwide.

And best of all, when you compare the LYNX price with the total cost of bringing any other modem on line, it could save you **OVER $100.**

The new LYNX. It's the latest innovation from the people who made data communications affordable for TRS-80 Model I and Apple II users. It's the best first step you can take in expanding. It makes your Model III a whole new animal.

STILL AVAILABLE—LYNX for TRS-80 Model I. Price: **$279.95.**

ESI LYNX ✓278

123 LOCUST STREET LANCASTER, PENNSYLVANIA 17602
Phone 717/291-1116

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation.

Call one of these LYNX handlers today—TOLL FREE:

ADVENTURE INTERNATIONAL
Longwood, FL
800/327-7172

THE PROGRAM STORE
Washington, DC
800/424-2738

ADVANCED COMPUTER PRODUCTS
Irvine, CA
800/854-8241

STEVENS RADIO SHACK
Phoenixville, PA
800/345-6279

TSE HARDSIDE
Milford, NH
800/258-1790

SIMUTEK
Tucson, AZ
800/528-1149

VISA

master charge

"I decided I would need to add a time delay to let the printer cool off between lines. The original screen print time was about two and a half minutes long. My first time delay extended the total time to five minutes."

```
7F6A CDDF7F    01700         CALL   SPOT
7F6D CB4E      01710         BIT    1,(HL)            ;TEST OTHER BLOCK
7F6F CDDF7F    01720         CALL   SPOT
7F72 23        01730         INC    HL                ;MOVE ALONG
7F73 10F3      01740         DJNZ   TOPLIN            ;FINISH THE TOP LINE
7F75 B7        01750         OR     A                 ;RESET CARRY FLAG
7F76 ED52      01760         SBC    HL,DE             ;PUT HL BACK TO START
7F78 CDC17F    01770         CALL   BLANKS            ;FILL OUT LINE
7F7B CDB37F    01780         CALL   TRSMIT            ;REPEAT FOR OTHER BITS
7F7E CDEE7F    01790         CALL   DELAY
7F81 0640      01800         LD     B,64
7F83 CB56      01810 MIDLIN  BIT    2,(HL)
7F85 CDDF7F    01820         CALL   SPOT
7F88 CB5E      01830         BIT    3,(HL)
7F8A CDDF7F    01840         CALL   SPOT
7F8D 23        01850         INC    HL
7F8E 10F3      01860         DJNZ   MIDLIN
7F90 B7        01870         OR     A
7F91 ED52      01880         SBC    HL,DE
7F93 CDC17F    01890         CALL   BLANKS
7F96 CDB37F    01900         CALL   TRSMIT
7F99 CDEE7F    01910         CALL   DELAY             ;SO PRINTER CAN COOL OFF
7F9C 0640      01920         LD     B,64
7F9E CB66      01930 BOTLIN  BIT    4,(HL)
7FA0 CDDF7F    01940         CALL   SPOT
7FA3 CB6E      01950         BIT    5,(HL)
7FA5 CDDF7F    01960         CALL   SPOT
7FA8 23        01970         INC    HL
7FA9 10F3      01980         DJNZ   BOTLIN
7FAB CDC17F    01990         CALL   BLANKS
7FAE C1        02000         POP    BC                ;GET LINE COUNTER
7FAF 10AB      02010         DJNZ   LOOP              ;FINISH OTHER LINES
7FB1 D9        02020         EXX
7FB2 C9        02030         RET
7FB3 3E1B      02040 TRSMIT  LD     A,27              ;ESCAPE
7FB5 CDD47F    02050         CALL   PRTOUT
7FB8 3E63      02060         LD     A,99              ;TRAMSMIT GRAPHICAL DATA
7FBA CDD47F    02070         CALL   PRTOUT
7FBD CDCA7F    02080         CALL   BLANK
7FC0 C9        02090         RET
7FC1 CDCA7F    02100 BLANKS  CALL   BLANK
7FC4 3E0D      02110         LD     A,13              ;CAUSES LINE FEED
7FC6 CDD47F    02120         CALL   PRTOUT
7FC9 C9        02130         RET
7FCA 0640      02140 BLANK   LD     B,64              ;NUMBER OF BLANKS
7FCC 3E80      02150         LD     A,128             ;BLANK
7FCE CDD47F    02160 BLINK   CALL   PRTOUT
7FD1 10FB      02170         DJNZ   BLINK
7FD3 C9        02180         RET
7FD4 E5        02190 PRTOUT  PUSH   HL                ;SAVE HL
7FD5 21E837    02200         LD     HL,37E8H          ;LP POINTER
7FD8 CB7E      02210 PRTLP8  BIT    7,(HL)            ;BIT 7 ON MEANS BUSY
7FDA 20FC      02220         JR     NZ,PRTLP8         ;THAT'S NOT MY LABEL
7FDC 77        02230         LD     (HL),A            ;LP READY, SO PRINT
7FDD E1        02240         POP    HL                ;GET BACK OLD HL
7FDE C9        02250         RET                      ;THAT WAS QUICK
7FDF 2809      02260 SPOT    JR     Z,NOSPOT          ;Z FLAG IS PASSED
7FE1 3E7F      02270         LD     A,7FH             ;ALL 7 BITS MAKE 1 BLOCK
7FE3 CDD47F    02280 TWOLIN  CALL   PRTOUT            ;MAKE TWWO LINE
7FE6 CDD47F    02290         CALL   PRTOUT
7FE9 C9        02310         RET
7FEA 3E80      02320 NOSPOT  LD     A,128             ;BLANK
7FEC 18F5      02330         JR     TWOLIN
7FEE 0602      02340 DELAY   LD     B,2               ;NUMBER OF TIME LOOPS
7FF0 C5        02350 OUTLUP  PUSH   BC                ;SAVE OUTSIDE COUNTER
7FF1 010000    02360         LD     BC,0              ;DO 65536 LOOPS
7FF4 0B        02370 TICTOC  DEC    BC
7FF5 78        02380         LD     A,B               ;SEE IF BC IS ZERO AGAIN
7FF6 B1        02390         OR     C
7FF7 20FB      02400         JR     NZ,TICTOC
7FF9 C1        02410         POP    BC
7FFA 10F4      02420         DJNZ   OUTLUP
7FFC C9        02430         RET    ;ADDRESS SHOULD BE 7FFFH OR LESS
06CC           02440         END    6CCH              ;ENTRY TO LEVEL II BASIC
00000 TOTAL ERRORS
```

"You should be able to easily make your own shapes by drawing them first on paper and assigning coordinates to each point."

```
3000 REM  DATA SET NO. 1      SIMPLE TISSUE BOX
3010 DATA 0,0,0,12,0,0
3020 DATA 12,0,0,12,25,0
3030 DATA 12,25,0,0,25,0
3040 DATA 0,25,0,0,0,0
3050 DATA 0,0,5,12,0,5
3060 DATA 12,0,5,12,25,5
3070 DATA 12,25,5,0,25,5
3080 DATA 0,25,5,0,0,5
3090 DATA 0,0,0,0,0,5
3100 DATA 12,0,0,12,0,5
3110 DATA 12,25,0,12,25,5
3120 DATA 0,25,0,0,25,5
3130 DATA 2,12.5,5,6,2,5
3140 DATA 6,2,5,10,12.5,5
3150 DATA 10,12.5,5,6,23,5
3160 DATA 6,23,5,2,12.5,5
3999 DATA 1000,,,,,
```

Data Set 1. Tissue Box

Line routine. If you modify the machine language program, remember to adjust the addresses in lines 1600–1630. These addresses should be one higher than the values of the corresponding labels in the machine language program.

The Third Dimension

Program Listing 4 will draw representations of three-dimensional objects. I have included data for a tissue box, a bolt and a hat.

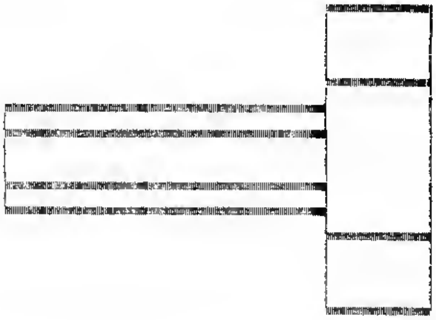


Fig. 6. This is a simple side view of the bolt.

The program will read data from the end of the program and then present a menu. The menu has five functions. The first three rotate the object. Fig. 2 shows the direction of each axis and the direction of rotation

```
3000 REM  DATA SET NO. 2     BOLT
3010 DATA 0,100,0,87,50,0
3020 DATA 87,50,0,87,-50,0
3030 DATA 87,-50,0,0,-100,0
3040 DATA 0,-100,0,-87,-50,0
3050 DATA -87,-50,0,-87,50,0
3060 DATA -87,50,0,0,100,0
3070 DATA 0,100,600,87,50,600
3080 DATA 87,50,600,87,-50,600
3090 DATA 87,-50,600,0,-100,600
3100 DATA 0,-100,600,-87,-50,600
3110 DATA -87,-50,600,-87,50,600
3120 DATA -87,50,600,0,100,600
3130 DATA 0,100,0,0,100,600
3140 DATA 87,50,0,87,50,600
3150 DATA 87,-50,0,87,-50,600
3160 DATA 0,-100,0,0,-100,600
3170 DATA -87,-50,0,-87,-50,600
3180 DATA -87,50,0,-87,50,600
3190 DATA 0,100,600,0,300,600
3200 DATA 87,50,600,260,150,600
3210 DATA 87,-50,600,260,-150,600
3220 DATA 0,-100,600,0,-300,600
3230 DATA -87,-50,600,-260,-150,600
3240 DATA -87,50,600,-260,150,600
3250 DATA 0,300,600,260,150,600
3260 DATA 260,150,600,260,-150,600
3270 DATA 260,-150,600,0,-300,600
3280 DATA 0,-300,600,-260,-150,600
3290 DATA -260,-150,600,-260,150,600
3300 DATA -260,150,600,0,300,600
3310 DATA 0,300,800,260,150,800
3320 DATA 260,150,800,260,-150,800
3330 DATA 260,-150,800,0,-300,800
3340 DATA 0,-300,800,-260,-150,800
3350 DATA -260,-150,800,-260,150,800
3360 DATA -260,150,800,0,300,800
3370 DATA 0,300,600,0,300,800
3380 DATA 260,150,600,260,150,800
3390 DATA 260,-150,600,260,-150,800
3400 DATA 0,-300,600,0,-300,800
3410 DATA -260,-150,600,-260,-150,800
3420 DATA -260,150,600,-260,150,800
3999 DATA 1000,,,,,
```

Data Set 2. Bolt

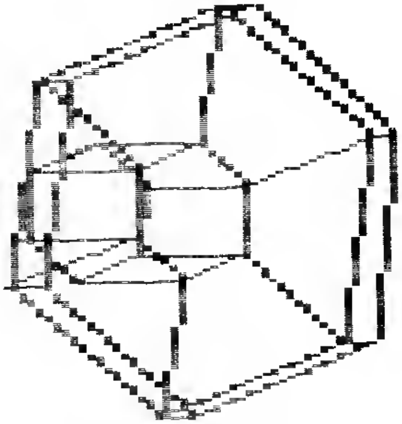


Fig. 7. This is from a point almost directly above or below the bolt.

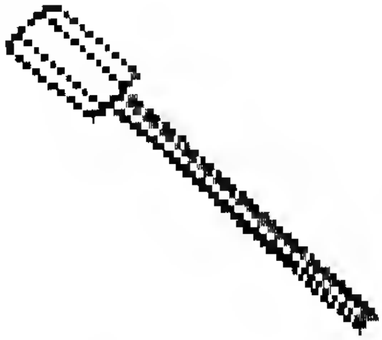


Fig. 8. The bolt can be stretched into a screwdriver for hexagonal screws.

"The manual said that the printer was getting too hot. I said, ' How can it get too hot on the first line? ' The darn manual didn't even reply."

about each axis. Function four allows the user to stretch the object in any or all directions. For example, if you wanted to make the object twice as wide, you could do this by doubling the Y values.

The display function clears the screen and draws the object. After the object has been drawn, the program will accept only three letters. M returns to the menu, R reverses the video, and S saves the display on the printer. The program automatically adjusts the data so that the object will fill the screen.

The program uses some fairly complicated mathematics. Line 50 reserves space for the data to draw one hundred lines. Lines 60–90 read the data. The variable N counts data statements. The end of the data is signified by a value of 1000 for X1. Do not use this value except at the end of your data, or the program will miss part of your design. Lines 110–180 print the menu and distribute control to the selected function. If the function is a rotation, line 190 asks for the number of degrees the object will be rotated. Control is then distributed to the individual function.

Before the actual rotation is started, the values of all points are checked to see if any are zero. If a variable is equal to zero, all points are shifted a tiny amount to avoid the possibility of a division by zero error. This check is done in line 200.

Lines 220–290 rotate the figure around the X axis. The angle of each point is computed in line 230. The rotation angle is converted to radians and added to the previous angle of the point. The distance from the point to the X axis is computed in line 250. This distance is used as the radius in the polar coordinate form. Once the program has computed the new angle and the radius, the program has to convert the polar form back to Cartesian coordinates, which is done using simple tranformation formulas: $Y = R\cos(T)$, $Z = R\sin(T)$. This process is repeated for all points.

The stretch function is much simpler. It asks for factors to multiply the X, Y and Z directions by, and then loops through and multiplies each point by these factors. Zero will not be accepted as a factor because zero would completely flatten the object.

The display section is in lines 600–790. Line 600 clears the screen with the special Field command. The next section of the program sorts through the data to find the largest values for each point. The smallest value is then subtracted from the largest value to find a scaling factor which will be used to

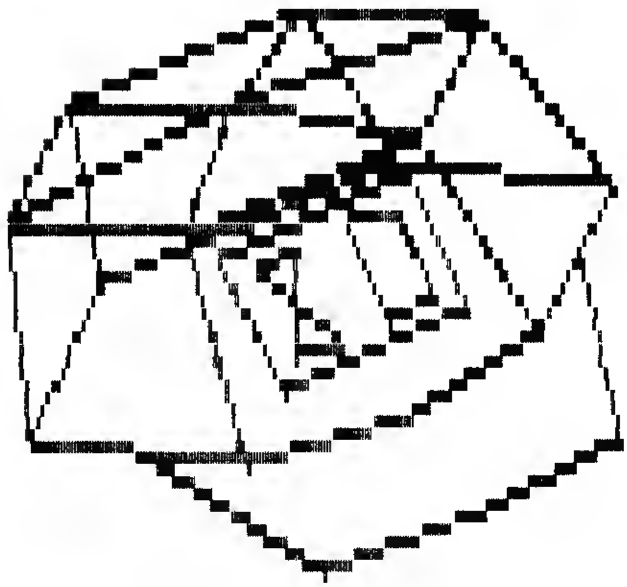


Fig. 9. This is a front view of the 80 Microcomputing *hat.*

Data Set 3: Hat with visor

```
3010 DATA 0,100,10,0,0,0
3020 DATA 87,50,10,0,0,0
3030 DATA 87,-50,10,0,0,0
3040 DATA 0,-100,10,0,0,0
3050 DATA -87,-50,10,0,0,0
3060 DATA -87,50,10,0,0,0
3070 DATA 0,100,10,0,125,100
3080 DATA 87,50,10,108,63,100
3090 DATA 87,-50,10,108,-63,100
3100 DATA 0,-100,10,0,-125,100
3110 DATA -87,-50,10,-108,-63,100
3120 DATA -87,50,10,-108,63,100
3130 DATA 0,100,10,87,50,10
3140 DATA 87,50,10,87,-50,10
3150 DATA 87,-50,10,0,-100,10
3160 DATA 0,-100,10,-87,-50,10
3170 DATA -87,-50,10,-87,50,10
3180 DATA -87,50,10,0,100,10
3190 DATA 0,125,100,108,63,100
3200 DATA 108,63,100,125,0,100
3210 DATA 125,0,100,108,-63,100
3220 DATA 108,-63,100,0,-125,100
3230 DATA 0,-125,100,-108,-63,100
3240 DATA -108,-63,100,-108,63,100
3250 DATA -108,63,100,0,125,100
3260 DATA 59,94,100,200,63,115     :REM NEXT 4 LINE ARE THE VISOR
3270 DATA 200,63,115,225,0,110
3280 DATA 225,0,110,200,-63,115
3290 DATA 200,-63,115,59,-94,100
3300 DATA 89,40,20,103,40,80      :REM REST OF THE DATA IS THE LABEL
3310 DATA 103,40,80,103,-40,80
3320 DATA 103,-40,80,89,-40,20
3330 DATA 89,-40,20,89,40,20
3340 DATA 92,30,30,92,10,30
3350 DATA 92,10,30,101,30,70
3360 DATA 101,30,70,101,10,70
3370 DATA 101,10,70,92,30,30
3380 DATA 92,-10,30,92,-30,30
3390 DATA 92,-30,30,101,-30,70
3400 DATA 101,-30,70,101,-10,70
3410 DATA 101,-10,70,92,-10,30
3999 DATA 1000,,,,,
```

expand the image to fill the screen. Line 690 finds the center of the image.

Lines 700–740 draw the object. The coordinates of the object are multiplied by scaling factors which compensate for the shape of the graphic blocks. For this reason, no matter how an object is rotated, every angle will remain correct.

Lines 750–790 scan the keyboard for an M, R or S. If you do not have a printer capable of printing graphical data, you can eliminate line 780. The reverse video routine was included in case somebody wanted to view the object on a white background.

The routine between lines 1500 and 1680 is the line drawing program.

Data set number one is for a simple tissue box. The data in each line is in the order X1, Y1, Z1, X2, Y2, Z2. Line number 3999 must be included at the end of the data. Fig. 5 was created by stretching the tissue box

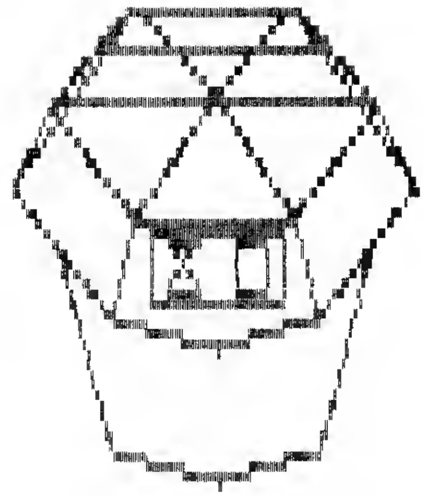


Fig. 10. This is a view from a 45 degree angle above the hat.

Program Listing 4

```
15 DEFINT Q,B
20 CLS
30 PRINT"READING DATA"
40 PI=3.1415926
50 DIM X(100,2),Y(100,2),Z(100,2)
60 N=1
70 READ X(N,1),Y(N,1),Z(N,1),X(N,2),Y(N,2),Z(N,2)
80 IF X(N,1) <> 1000 THEN N=N+1:GOTO70
90 N=N-1
100 PRINTN;"DATA LINES WERE READ"
110 PRINT"1 = ROTATE AROUND X AXIS"
120 PRINT"2 = ROTATE AROUND Y AXIS"
130 PRINT"3 = ROTATE AROUND Z AXIS"
140 PRINT"4 = STRETCH OR SHRINK"
150 PRINT"5 = DISPLAY"
160 INPUT"ENTER FUNCTION NUMBER";F
170 IFF<1ORF>5THEN160
180 ONFGOTO190,190,190,500,600
190 INPUT"ENTER THE NUMBER OF DEGREES TO ROTATE";D
200 FORQ=1TON:FORB=1TO2:IFX(Q,B)=0 OR Y(Q,B)=0 OR Z(Q,B
   )=0 THEN900 ELSE NEXT:NEXT
210 ONFGOTO220,300,380
220 FORQ=1TON:FORB=1TO2
230 T=ATN(Z(Q,B)/Y(Q,B))+D*PI/180
240 IFY(Q,B)<0THENT=T+PI
250 R=SQR(Z(Q,B)*Z(Q,B)+Y(Q,B)*Y(Q,B))
260 Y(Q,B)=COS(T)*R
270 Z(Q,B)=SIN(T)*R
280 NEXT:NEXT
290 GOTO110
300 FORQ=1TON:FORB=1TO2
310 T=ATN(Z(Q,B)/X(Q,B))+D*PI/180
320 IFX(Q,B)<0 THENT=T+PI
330 R=SQR(Z(Q,B)*Z(Q,B)+X(Q,B)*X(Q,B))
340 X(Q,B)=COS(T)*R
350 Z(Q,B)=SIN(T)*R
360 NEXT:NEXT
370 GOTO110
380 FORQ=1TON:FORB=1TO2
390 T=ATN(Y(Q,B)/X(Q,B))+D*PI/180
400 IFX(Q,B)<0THENT=T+PI
410 R=SQR(Y(Q,B)*Y(Q,B)+X(Q,B)*X(Q,B))
420 X(Q,B)=COS(T)*R
430 Y(Q,B)=SIN(T)*R
440 NEXT:NEXT
```

Program continues

```
450 GOTO110
500 INPUT"ENTER A FACTOR FOR X, Y AND Z IN THAT ORDER";
     X,Y,Z
510 IF X=0 OR Y=0 OR Z=0 THEN PRINT"ZERO IS INVALID.  T
     RY AGAIN":GOTO500
520 FORQ=1TON:FORB=1TO2
530 X(Q,B)=X(Q,B)*X:Y(Q,B)=Y(Q,B)*Y:Z(Q,B)=Z(Q,B)*Z:NEX
     T:NEXT
540 GOTO110
600 FIELD
610 HY=Y(1,1):LY=HY:HZ=Z(1,1):LZ=HZ
620 FORQ=1TON:FORB=1TO2
630 IF Y(Q,B)>HY THEN HY=Y(Q,B)
640 IF Y(Q,B)<LY THEN LY=Y(Q,B)
650 IF Z(Q,B)>HZ THEN HZ=Z(Q,B)
660 IF Z(Q,B)<LZ THEN LZ=Z(Q,B)
670 NEXT:NEXT
680 W=HY-LY:IFHZ-LZ>WTHENW=HZ-LZ
690 SY=-(HY+LY)/2:SZ=-(HZ+LZ)/2
700 FORQ=1TON
710 X1=63.5+(Y(Q,1)+SY)/W*2.4*48:Y1=23.5-(Z(Q,1)+SZ)/W*
     48
720 X2=63.5+(Y(Q,2)+SY)/W*2.4*48:Y2=23.5-(Z(Q,2)+SZ)/W*
     48
730 GOSUB1500
740 NEXT
750 Z$=INKEY$:IFZ$=""THEN750
760 IFZ$="M"THEN110
770 IFZ$="R"THENRSET:GOTO750
780 IFZ$="S"THENSAVE:GOTO110
790 GOTO750
900 S=.001:FORQ=1TON:FORB=1TO2:X(Q,B)=X(Q,B)+S:Y(Q,B)=Y
     (Q,B)+S:Z(Q,B)=Z(Q,B)+S:NEXT:NEXT:GOTO200
1500 REM LINE ROUTINE
1510 X1=FIX(X1):X2=FIX(X2):Y1=FIX(Y1):Y2=FIX(Y2)
1520 IF X1 < 0 OR X1 > 127 THEN PRINT"X1 ILLEGAL":END
1530 IF X2 < 0 OR X2 > 127 THEN PRINT"X2 ILLEGAL":END
1540 IF Y1 < 0 OR Y1 > 47 THEN PRINT"Y1 ILLEGAL":END
1550 IF Y2 < 0 OR Y2 > 47 THEN PRINT"Y2 ILLEGAL":END
1560 IF INT(X1)=INT(X2) AND INT(Y1)=INT(Y2) THEN SET(X1
     ,Y1):RETURN
1600 POKE32331,X1
1610 POKE32333,Y1
1620 POKE32335,X2
1630 POKE32337,Y2
1660 LINE
1680 RETURN
```

three times in the Z direction.

The second data set draws a bolt. Most readers will notice that the bolt has a hexagonal shaft rather than a round one; well, let's see *you* try to draw a circle with straight lines. Fig. 8 was created by stretching the bolt into a screwdriver for hexagonal screws. A top view of the bolt looks like a hexagonal wheel which can be put on a cart to hold all these special parts we are designing. I hope that everyone realizes the tremendous value of a wheel that won't roll away.

The third data set defines a special *80 Microcomputing* hat. This is the only object for which its direction can be determined. For example, if the label has an 80 on it, you are looking at the front, but if the label has an 08 on it, you are looking at the back. The hat is made of the same material as the transparent tissue box and the transparent bolt. The hat will be upside down when the data is first read, but it can be turned over by rotating it 180 degrees around the X axis.

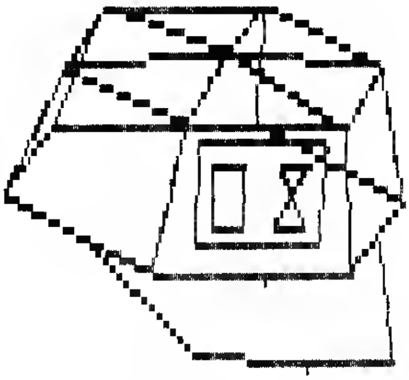


Fig. 11. This is a view from the rear of the hat looking up through the bottom.

You should be able to easily make your own shapes by drawing them first on paper and assigning coordinates to each point. After coordinates have been assigned, type in one data line for each line in the drawing. ■

What has nine lives, three forms, multiple faces and a price tag that almost disappears?

The Magical Microline 80 Printer

It's magic! Well, almost. The Microline 80 will run all day at 80 cps with no duty cycle limitations. The head is warranted for 200,000,000 characters. That translates to over nine years on your TRS-80,™ APPLE® or other small computer.

Want to change forms? The magical Microline 80 is three printers disguised as one. There is a whisper-quiet rubber platen for cut sheets and roll paper, pins on nine inch centers for pin feed stock and optional snap-on tractors that adjust to suit all your other forms. The Microline 80 also saves paper by letting you tear off as close as one inch from the last print line.

Want to change your image? The magical Microline 80 really does tricks. It prints upper and lower case, condensed and double width characters and block graphics for charts, graphs and diagrams.

The Microline 80 is not a toy. With two motors, a rugged cast aluminum base and a head you never have to throw away, the Microline 80 is built to handle the most demanding business applications.

Which brings us to the biggest magic of all, the price tag, the one that almost disappears. If we're not the lowest, we are so close that it doesn't matter. There are stocking Microline distributors throughout the country. Call or write today for the name of the one near you and the price of the Magical Microline 80.

OKIDATA

Okidata Corporation, 111 Gaither Drive, Mount Laurel, New Jersey 08054 609-235-2600

Okidata is a subsidiary of Oki Electric Industry Company, Ltd.

✓245

80 Microcomputing's Buyers Guide To Printers

Though our charts are far from complete, 80 Microcomputing *started with an exhaustive list of manufacturers and OEM's in hopes of bringing you the most complete picture of the printer industry. The editors tried to eliminate all the intelligent terminals and those keyboard models that we felt were not comparable on a price/feature basis. The following chart lists all of those manufacturers who were thoughtful enough to reply.*

	Company	Model	Char./Sec.	Lines/Min.	Lines/Inch	Char./Inch	Char./Line
$0–$550	Coosol Inc.	101B-80E	160	60	6	11-13	88
	Coosol Inc.	101B-48E	100	50	6	14	48
	Dip Inc.	81	100	60	6	5-10	40/80
	Printel Inc	Sidewinder	30	—	9	9.6	132
	Radio Shack	LP7	30	—	—	—	80
	Radio Shack	QP2	64	120	6	18/9	32/16
$551–$700	Base 2, Inc.	800	100	60	6	11	64/72/80 96/120/132
	Comprint	912-S	225	170	5.8	11	80
	Comprint	912-P	225	170	5.8	11	80
	Comprint	912-GPP	225	170	5.8	11	80
	Coosol Inc.	102B-133E	160	120	6	11-3	132
	Heath Computer	H-14	45	—	6/8	—	80/96/132
	Okidata Corp.	Microline 80	80	86-28	6/8	5/10/16.5	20/40/80

Interface[1]
P = Parallel
R = rs-232
T = 20mA
I = IEEE

Baud Rates[2]
A = 50–9600 baud
B = 110–9600 baud
C = 399–9600 baud
D = 110–4800 baud
E = 110–1200 baud
F = 600 baud
G = 50–19200 baud
H = 15–9600 baud
I = 75–9600 baud

Form Feed[3]
T = tractor
P = pin
F = friction
SS = single sheet
DS = double sheet
A = auto sheet
C = cutsheet feeder
S = adjustable width sprocket

The printer industry is one of the most aggressive segments of the microcomputer peripherals market.

According to one study by Dataquest, a Cupertino, CA, market research firm, the total printer market is expected to grow from the 425,000 units shipped in 1980 to 760,000 units in 1983. By 1983 the total value of the market is expected to exceed $2 billion.

> *"The most competitive slice of the printer market falls under the $1,000 bracket."*

The most competitive slice of this market, according to *Electronics News*, an industry trade journal, is the under $1,000 bracket. Supported largely by consumer microcomputer owners, this market is expected to more than triple over the next three years, growing from 65,000 units shipped in 1980 to 275,000 units in 1983. The total dollar value of the market will grow by less than 300 percent over this same time period because of the expected price breaks and mass market technologies that will prevail.

Dataquest set the dollar figure at $52 million last year and pegs it at $137.5 million for 1983. This under $1,000 bracket includes both impact and non-impact printers.

A Creative Strategies International survey which grouped so-

The DIP Inc. Model 81 Dot Matrix Printer

Interface[1]	Baud Rates[2]	Paper Width	Form Feed[3]	Technology[4]	Head Type[5]	Dot Matrix Format	Multi-pass	Bidirectional	Upper and Lowercase	Char. Set[6] (number of ASCII char.)	Print Head Life (millions of characters)	Cable Included	Weight (lbs.)	Prices	
P,R	B	1-10	T	I	DM	5×7 to 10×14	Y	N	Y	96	—	N	20	$545	$0–$550
P,R	B	4	F	I	DM	5×7 to 10×14	N	N	Y	96	—	N	12	$355	
P Opt.-R,T	E	8.5	F	I	—	—	N	Y	Y	96	100	Y	12	$499	
P	—	14	F	T	DM	5×5	N	N	N	64	70	N	2.5	$199	
P,R	F	9.5	T	I	DM	5×7	N	N	Y	96G	100	N	8.6	$399	
P,R	F	2 3/8	T	E	DM	5×7	N	—	Y	96	30	N	—	$219	
P,R,T	I	9.5	T,F	I	DM	5×7	N	Y	Y	96	100	N	15	$699	$551–$700
R,T	D	8.5	F	E	DM	9×12	N	N	Y	96	100	Y	15	$699	
I	D	8.5	F	E	DM	9×12	N	N	Y	96	100	Y	15	$660	
P	D	8.5	F	E	DM	9×12	N	N	Y	96	100	Y	15	$660	
P,R,T	B	1-10	T,P	I	DM	5×7 10×14	Y	Y	Y	128G	—	N	20	$595	
R,T	D	2.5-9.5	S	I	DM	5×7	N	N	Y	96	100	Y	30	$695	
P Opt.-R,T	B	4-9	P,F Opt.-T	I	DM	9×7	—	N	Y	96+64G	200	N	14	$599	

Technology[4]
- I = impact
- T = thermal
- E = electrostatic

Head Type[5]
- DM = dot matrix
- D = daisy
- T = thimble

Character Set[6]
- G = TRS-80 graphic characters
- I = international characters
- F = foreign sets
- OC = optional characters

*
- HS = high speed
- LQ = letter quality
- P = proportional spacing
- M = monospeed

LPV Graphics Executed in Expanded Mode from Radio Shack

Price	Company	Model	Char./Sec.	Lines/Min.	Lines/Inch	Char./Inch	Char./Line
$701–$900	Base 2, Inc.	850	100	60	6	11	64/72/80 96/120/132
	Centronics Data Computer Corp.	730	100	30	6	5/10/16.5	40/80/132
	DIP Inc.	84	100	60	6/8	5-16.5	40/48/66 96/132
	DIP Inc.	85	100	60	6/8	5.16.5	40/48/66 96/132
	Integral Data Systems	445	198	42	6/8	4.2/5/6/8.4 10/12/16.8	80
	Microtek Inc.	MT80	125	60	6	5/10/16.5	40/80/132
	Okidata	Microline 82	80	137-50	6/8	5/10/16.5	20/40/80
	Radio Shack	LP2	100	31	—	—	8/132
$901–$1200	Anadex Inc.	DP-8000	112	84	6	5/10	80
	Centronics Data Computer Corp.	737	50	21	6	5/10/16.5	40/80/132
	Centronics Data Computer Corp.	739	100 M* 80 P*	—	6	10/16.7	80/132
	Dataroyal Inc.	IPS-5000	125	—	6 Opt.-6/8	10	80
	Dataroyal Inc.	IPS-5000A	150	—	6 Opt.-6/8	10	80
	Olivetti Peripheral Equipment	TH 240	320	240	6	10	80
	Radio Shack	LP4	50	22	—	—	80/130
	Radio Shack	LP6	100	33	—	—	132
$1201–$1600	Anadex Inc.	DP-9000	112	84	6	5/10	106
	Anadex Inc.	DP-9001	150/200	—	6/8	5/6/6.7 10/12/13.3	134
	Centronics	779	60-100	21-90	6	10/16.5	80-132
	Dataroyal Inc.	IPS-5000	125	—	6 Opt.-6/8	10	136
	Dataroyal Inc.	IPS-5000A	150	P	6 Opt.-6/8	10	136
	Infoscribe Inc.	500	150	—	6/8	10/12/16.5	136/163/224
	Infoscribe Inc.	1000	180	—	6/8	10/12/16.5	136/163/224
	Integral Data Systems	460	160	60	6/8	5/6/8.4 10/12/16.8	80
	Okidata Inc.	Microline 83	120	212-76	6/8	5/10/16.5	132
	Teletype Corp.	4220	30	—	—	10/13	80/132

Interface[1]
P = Parallel
R = rs-232
T = 20mA
I = IEEE

Baud Rates[2]
A = 50–9600 baud
B = 110–9600 baud
C = 399–9600 baud
D = 110–4800 baud
E = 110–1200 baud
F = 600 baud
G = 50–19200 baud
H = 15–9600 baud
I = 75–9600 baud

Form Feed[3]
T = tractor
P = pin
F = friction
SS = single sheet
DS = double sheet
A = auto sheet
C = cutsheet feeder
S = adjustable width sprocket

LPV Graphics Executed in Expanded Condensed Mode from Radio Shack

Interface[1]	Baud Rates[2]	Paper Width	Form Feed[3]	Technology[4]	Head Type[5]	Dot Matrix Format	Multi-pass	Bidirectional	Upper and Lowercase	Char. Set[6] (number of ASCII char.)	Print Head Life (millions of characters)	Cable Included	Weight (lbs.)	Prices
$701–$900														
P,I,R,T	I	9.5	T,F	I	DM	5×7	N	Y	Y	96	100	N	15	$799
P,R	A	9	P,F	I	DM	7×7	N	N	Y	96	50	N	10	$795
P,R,T	E	9.5	T,F	I	—	—	N	Y	Y	96G	100	Y	12	$795
P,R,T	B	9.5	T,F	I	—	—	Y	Y	Y	96G	100	Y	12.6	$895
P,R	E	1.75-9.5	T	I	DM	7×7 8×7	N	N	Y	96	250	N	16.5	$795
P,R	B	1-9.5	P	I	DM	9×7	N	Y	Y	96	100	N	22	$795
P,R,T	B	4-9	P,F Opt.-T	I	DM	9×7	—	Y	Y	96+64G	200	N	17.6	$799
P	—	9.5	P	I	DM	5×7	N	N	Y	96	100	N	10	$799
$901–$1200														
P,R,T	B	9.5	P	I	DM	9×7	N	Y	Y	95	100	N	20	$1225
P,R	G	9	P,F	I	DM	N×7 P* 7×8 M*	N	N	Y	96	250	N	12	$995
P,R	—	8.5	P,SS	I	DM	N×7 P* 7×8 M*	—	Y	Y	96I	—	—	11.88	Less than $1000
P,Opt-T	B	3-11.5	P,F	I	DM	9×9	N	Y	Y	96	—	N	30	$1100
P,R Opt.-T	B	3-11.5	P,F	I	DM	9×9	N	Y	Y	96,Opt.-I	—	N	30	$1160
P,R,T	H	8.5	F	T	DM	5×7	N	Y	Y	96	20	N	16.5	$1129
P	—	9.5	P	I	DM	N×7	N	N	Y	96	100	N	12	$999
P	—	14 7/8	T or F	I	DM	7×9	N	Y	Y	96 241,+G	100	N	28	$1160
$1201–$1600														
P,R,T	B	9.5	P	I	DM	9×7	N	Y	Y	95	100	N	34	$1550
P,R,T	A	9.5	T	I	DM	9×7 7×9	N	Y	Y	96	150	N	34	$1550
P	—	12.1	T	I	DM	5×7	N	N	N	64	250	N	45	$1485
P, Opt.-T	B	3-15	P,T	I	DM	9×9	N	Y	Y	96	—	N	35	$1295
P,R Opt.-T	B	3-15	P,T	I	DM	9×9	N	Y	Y	96,F	—	N	35	$1345
P,R,T	B	1.5-16	T	I	DM	9×9	N	Y	Y	96	300	N	32	$1395
P,R,T	B	1.5-16	T	I	DM	9×9	N	Y	Y	96	300	N	34	$1595
P,R	C	1.75-9.5	T	I	DM	24×9 48×9	N	Y	Y	96	250	N	20.7	$1295
P,R Opt.-T	B	4-14.38	F, Opt.-T	I	DM	9×7	—	Y	Y	96+64G	200	N	28.7	$1260
R,T	—	—	P,F	I	DM	7×9	N	N	Y	96	200	N	32	$1558 Pin $1593 Fric

Technology[4]
I = impact
T = thermal
E = electrostatic

Head Type[5]
DM = dot matrix
D = daisy
T = thimble

Character Set[6]
G = TRS-80 graphic characters
I = international characters
F = foreign sets
OC = optional characters

*****
HS = high speed
LQ = letter quality
P = proportional spacing
M = monospeed

called low-cost printers together sets their 1980 market value at over $100 million for approximately 125,000 units sold. By 1985, Creative projects the figures at 475,000 units shipped for a total market value in excess of $300 million.

Each study has impact printers outperforming non-impact varieties. Two technologies are currently used in most printer manufacture, impact or non-impact. Impact employs the traditional method of printing whereby a character is pressed onto the paper via a ribbon. Non-impact methods employ thermal, electro-static, ink jet and laser techniques and usually are more expensive.

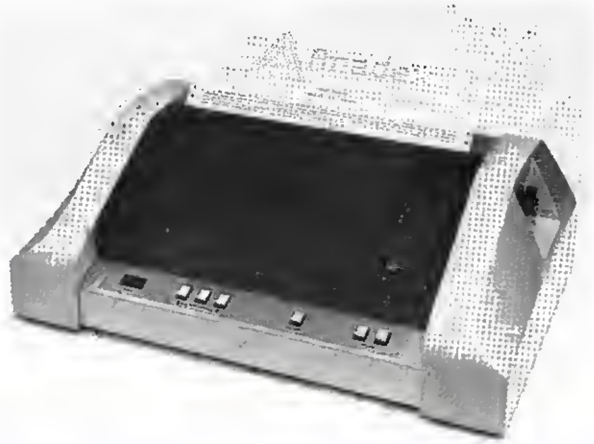
For most microcomputerists, impact printers are the norm. Of all the different types of impact printers available, the dot matrix is most popular with the microcomputerist and provides him with the greatest benefits at the lowest price.

The dot matrix printing technique is straightforward. A matrix, or rectangle, consisting of between 63 and 81 solenoid driven pins, impacts on to a sheet of paper through an inked ribbon. The matrix pin configuration is determined by the character the printer is to reproduce. The denser the matrix (the more pins), the better defined the characters will be.

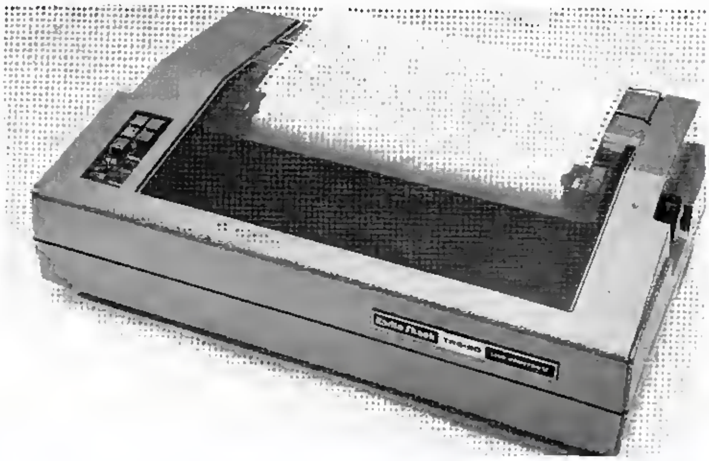
"This market is expected to triple over the next three years. . . ."

Dot matrix techniques have evolved to the point where print quality approaches that of a good typewriter.

Another impact technique prints fully-formed characters. The so-called daisywheel mechanism uses a revolving wheel to place print characters in front of a hammer. The wheel is cut into thin fingers which resemble the petals of a flower. Each finger has a character on it and a wide range of type styles can be had by simply changing print wheels.



Model DP-9500 from Anadex



Radio Shack's TRS-80 Line Printer V

	Company	Model	Char./Sec.	Lines/Min.	Lines/Inch	Char./Inch	Char./Line
$1601-$2000	Anadex Inc.	DP-9500	150/200	—	6/8	5/6/6.7 10/12/13.3	175
	Anadex Inc.	DP-9501	120/200	—	6/8	5/6.2/7.5/8.3 10/12.5/15/16.7	220
	Infoscribe Inc.	1500	360	—	6/8	10/12/16.5	136/163/224
	Integral Data Systems	560	160	60	6/8	5/6/8.4 10/12/16.8	132
	Olivetti Peripheral Equipment	DY-211	20	—	5/6	10/12/15 P*	132/158 198/114-198
	Radio Shack	LPV	160	60	—	—	112
	Radio Shack	DW2	43	20	—	—	132/195

Interface[1]
P = Parallel
R = rs-232
T = 20mA
I = IEEE

Baud Rates[2]
A = 50–9600 baud
B = 110–9600 baud
C = 399–9600 baud
D = 110–4800 baud
E = 110–1200 baud
F = 600 baud
G = 50–19200 baud
H = 15–9600 baud
I = 75–9600 baud

Form Feed[3]
T = tractor
P = pin
F = friction
SS = single sheet
DS = double sheet
A = auto sheet
C = cutsheet feeder
S = adjustable width sprocket

Print quality of daisywheel units is high, but because of the mechanics involved in rotating the wheel, the speed is not great. In addition, daisywheel units are usually more expensive than dot matrix units.

The most frequently seen type of non-impact printer is the thermal or heat sensitive unit. This printer uses a heated dot matrix head to reproduce characters on special, heat sensitive paper. Other configurations of this type of printer exist but, in all cases, special paper is required—a definite drawback when large amounts of copy are necessary.

Though fully formed character printers have long dominated the market, printing at speeds of up to 1100 lines per minute at the standard 132 characters across, their prices have kept them out of reach of the home user, or those customers without heavy word processing needs. In the under $1000 bracket that is causing much market activity, it is the influx of a number of reliable dot matrix printers that account for the projected growth rates.

In the low price market that is capturing the attention of the home

Sample Graphics from the new Centronics Model 739

The Centronics Model 737

$1601–$2000

	Interface[1]	Baud Rates[2]	Paper Width	Form Feed[3]	Technology[4]	Head Type[5]	Dot Matrix Format	Multi-pass	Bidirectional	Upper and Lowercase	Char. Set[6] (number of ASCII char.)	Print Head Life (millions of characters)	Cable Included	Weight (lbs.)	Prices
	P,R,T	A	16	T	I	DM	9×9 7×9	N	Y	Y	96	150	N	36	$1650
	P,R,T	A	16	T	I	DM	11×9 7×9	N	Y	Y	96	150	N	36	$1650
	P,R,T	B	1.5-16	T	I	DM	9×9	N	Y	Y	96	300	N	38	$1[illegible]95
	P,R	C	1.75-14.75	T	I	DM	24×9 48×9	N	Y	Y	96	250	N	20.7	$1695
	P,R	B	17	F Opt.—T,SS	I	D	—	N	Y	Y	96 Opt.-OC	10	N	36	$1900
	P	—	14 7/8	T	I	DM	7×9	N	Y	Y	96 241,30G	100	N	42	$1860
	P	—	15	F	I	D	—	N	N	Y	124	40	N	28	$1960

Technology[4]	Head Type[5]	Character Set[6]	*
I = impact	DM = dot matrix	G = TRS-80 graphic characters	HS = high speed
T = thermal	D = daisy	I = international characters	LQ = letter quality
E = electrostatic	T = thimble	F = foreign sets	P = proportional spacing
		OC = optional characters	M = monospeed

The NEC Spinwriter 5525

user, Japanese imports such as Epson, Okidata and Itoh are making their greatest advances.

Further, technology is also transforming our consumer habits. The increased speeds of dot matrix and the improved character shaping, available at a fraction of the price of traditional line printers with fully-formed—so-called letter quality—characters has chastened our judgment about the clarity of letter quality printers versus the dot matrix.

This activity can only be accelerated by IBM's addition of their 5225, a high-speed dot matrix printer manufactured by Dataproducts. As standards of readability are altered by the limits of technology, and technology continues to upgrade the performance of dot matrix printers by means of overlapping dots or double-pass methods, this market will see continued growth. Already small businessmen and home users are finding the dot matrix character sufficient for internal reporting and some word processing needs.

	Company	Model	Char./Sec.	Lines/Min.	Lines/Inch	Char./Inch	Char./Line
$2001–$3000	Dataproducts Corp.	D-50 MKI	50	—	3/4/6	10/12	—
	Dataproducts Corp.	D-50 RO Terminal	50	—	6/8	10/12	—
	Malibu Electronics Corp.	165	HS* LQ* 165 90	—	2/3/4/6 8/10/12	5/10	132
	Malibu Electronics Corp.	Dual Mode 200	HS* LQ* 165-250 42-70	—	2/3/4 6/8/10/12	10/12/17 5/6/8.5 E*	132
	Olivetti Peripheral Equipment	DY311	32	—	5/6	10/12/15 P*	150/180/225 125-225
Over $3000	Olivetti Peripheral Equipment	DY811	65	—	5/6	10/12/15 P*	150/180/225 247/128-225
	Olivetti Peripheral Equipment	DM 80/180	80/180	—	5/6	P* 10/12/15/16.6	150/180/225 247/128-225
	NEC Information Systems Inc.	5530	55	—	6/8	10/12	136/163
	NEC Information Systems Inc.	5510	55	—	6/8	10/12	136/163
	NEC Information Systems Inc.	5515	55	—	6/8	10/12	136/163
	NEC Information Systems Inc.	5520	55	—	6/8	10/12	136/163
	NEC Information Systems Inc.	5525	55	—	6/8	10/12	136/163

Interface[1]
P = Parallel
R = rs-232
T = 20mA
I = IEEE

Baud Rates[2]
A = 50–9600 baud
B = 110–9600 baud
C = 399–9600 baud
D = 110–4800 baud
E = 110–1200 baud
F = 600 baud
G = 50–19200 baud
H = 15–9600 baud
I = 75–9600 baud

Form Feed[3]
T = tractor
P = pin
F = friction
SS = single sheet
DS = double sheet
A = auto sheet
C = cutsheet feeder
S = adjustable width sprocket

Though high speed fully-formed character printers such as those manufactured by Qume and Diablo still lead the sales market, they are meeting competition from traditional typewriter manufacturers who are turning their attention to the printer market. Induced by the success of dot matrix printers costing less than $1,000, major and minor manufacturers of thimble, daisywheel, element and drum printers are hastening to introduce fully-formed character printers that run slower and cost far less.

"It is the influx of reliable dot matrix printers that account for projected growth rates . . "

Among those manufacturers are Pertec, Olympia and Olivetti. Pertec has introduced its Stylist 360, manufactured by its parent firm, Triumph Adler of Germany. It is a daisywheel that operates between 17 and 20 characters per second and will cost much less than the 45 to 55 character counterparts from Qume and Diablo. They will be a price-compatible alternative to current dot matrix printers. ■


The Olivetti DM80/180

Interface[1]	Baud Rates[2]	Paper Width	Form Feed[3]	Technology[4]	Head Type[5]	Dot Matrix Format	Multi-pass	Bidirectional	Upper and Lowercase	Char. Set[6] (number of ASCII char.)	Print Head Life (millions of characters)	Cable Included	Weight (lbs.)	Prices	
—	—	15	F Opt.-T,A	I	D	—	—	Y	Y	96	—	N	57.5	$2776	$2001–$3000
R,T	C	15	F Opt.-T,A	I	D	—	—	Y	Y	96	—	N	57.5	$2776	
P,R	A	2-15	T	I	DM	5×9 10×9	—	Y	Y	96	500	N	48	$2295	
P,R	B	2-15	T,F	I	DM	9×9 19×18	Y	Y	Y	96	300	N	40	$2995	
P,R,T	B	17 1/4	F Opt.T,SS,DS	I	D	—	N	Y	Y	96 Opt.-OC,G	10	N	39.5	$2850	
R,T	B	17 1/4	F Opt.T,SS,DS	I	D	—	N	Y	Y	96 Opt.-OC,G	10	N	79	$4280	Over $3000
R,T	B	17 1/4	F Opt.T,SS,DS	I	DM	8×7 16×32	N	Y	Y	96 Opt.-OC,G	—	N	79	$4280	
P	—	16	T,P,F,C	I	T	—	—	Y	Y	128	30	N	45.5	$3055	
R,T	E	16	T,P,F,C	I	T	—	—	Y	Y	128	30	Y	45.5	$3055	
R,T	E	16	T,P,F,C	I	T	—	—	Y	Y	128	30	Y	45.5	$3135	
R,T	E	16	T,P,F,C	I	T	—	—	Y	Y	128	30	Y	51	$3415	
R,T	E	16	T,P,F,C	I	T	—	—	Y	Y	128	30	Y	51	$3455	

Technology[4]
I = impact
T = thermal
E = electrostatic

Head Type[5]
DM = dot matrix
D = daisy
T = thimble

Character Set[6]
G = TRS-80 graphic characters
I = international characters
F = foreign sets
OC = optional characters

*****
HS = high speed
LQ = letter quality
P = proportional spacing
M = monospeed

The price of progress is innovation.

The Xerox 1740 And the Model II

Dr. James H. Nestor
39114 Route 303
Grafton, Ohio 44044

Nestor's Law states that Radio Shack will always have the hardware and/or software I need six months (or more) after I need it. That statement may earn a place of prominence with Murphy's Law, Parkinson's Law, and the Peter Principle. Think about it.

No Foreign Equipment

I have endured the tirades of Radio Shack personnel for more than two years, ever since I hung a Pertec drive on my Model I instead of waiting six months for a "legitimate" drive. They have preached to me about the importance of using only Radio Shack peripherals. I have even conceded that for many people it is wise to get everything from one source. Where the argument breaks down is when I have a specific need and they are unable to fill it.

As case in point, I cite Scripsit, Radio Shack's word processing program. It was released about two years after I bought Electric Pencil and modified my Model I for lowercase.

When I purchased a Model II the problem remained. In fact, I think it got worse. While Scripsit for Model I was now available, there was no version for Model II. Nor were the folks in Fort Worth even talking about a release date. Here I sat with the perfect machine for word processing: 80 × 24 display with true lower case, 1/2 megabyte of disk storage, full keyboard, and both parallel and serial printer ports. Not to mention reverse video and 64K of RAM! But, alas, there was no software and no letter quality printer. Phooey again!

Undaunted, I set out in search of counterfeit means to enjoy my machine. The search, though hectic, was worthwhile. In fact, I'm going to share my secrets with you in this very article.

Finding a Printer

If all else failed, I thought I might be able to write some word processing software. I would never presume to be able to build a printer. Therefore, I set out in search of a printer first. Finding one wasn't really all that tough. I excluded the various Selectrics because of their slow print speed and mechanical complexity.

That left three candidates: NEC Spinwriter, Qume Sprint, and Xerox 1740. All three are daisywheel types with a minimum of moving parts and excellent print quality. Although all three appear to be excellent printers, I settled on the Xerox 1740 because of availability of local service. The same machine is marketed through dealers as the Diablo 1640. I should point out that although Xerox manufactures both models, they will only service the Xerox 1740.

Getting the Printer to Work

I selected a 1740 RO (Receive Only) version because I didn't need another keyboard. The printer has an RS-232 serial interface. Since the Model II also has two RS-232 serial ports, I assumed it would be a plug-in installation. Not true.

Although the plug on the 1740 would fit either Port A or Port B, I could not get the Model II to talk to the printer. Or possibly, the printer wasn't listening to the Model II. If Xerox wouldn't service a Diablo printer, just maybe the 1740 refused to respond to so lowly a driver as the Model II. Panic!

Problem One: Hardware

At this point I assumed that I had a hardware problem and proceeded accordingly. I read and re-read the Xerox tech manual which came with the 1740. I reset all of the switches, and tried the Self Test (sounds Freudian). It worked. The printer produced several lines of perfect print. I read and re-read the Model II owner's manual.

Article Two of Nestor's Law says that "Adequate documentation, especially technical manuals, shall immediately follow discontinuation of any given model." The only item of value in the Model II manual was a pin-out diagram of serial ports A and B. I compared it to the diagram in the Xerox manual. They were identical. At first I thought that was terrific. They are both RS-232, so they should be identical. It even made sense, so I knew it had to be wrong.

Digging further into the respective manuals, I discovered the "rub". After studying the concept of handshaking, I concluded that the pin-outs should not be identical. The clear-to-send line of one device does not connect to the clear-to-send of the other. Rather, it goes to the request-to-send pin. I'll not keep you in suspense any longer. In order for the Xerox 1740 to work with the Model II, a few changes must be made in the wiring of the connecting cable.

Solution One: Hardware

The cable is soldered in place inside the printer, so the modifications must be made to the wires inside the connector on the cable end. Four wires in the plug must be switched, and a jumper must be soldered

Vista

Model V1000

. . . Looking Out For You.

Eight Inch Floppy Disk Drive Subsystem Model V1000

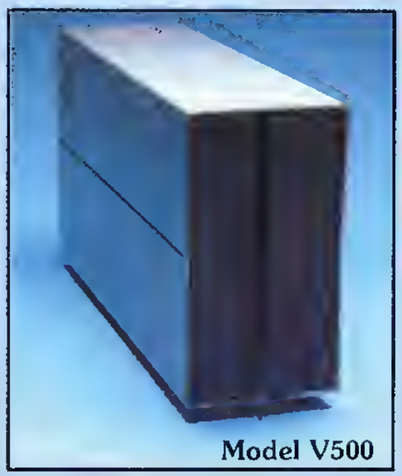
The V1000, Vista's sophisticated new disk drive subsystem, sets new standards for ease of access and use. Its innovative design permits disk drives to be mounted or removed quickly and easily for system reconfiguration or servicing.

Features:

- Deluxe chassis with internal slide allows easy access.
- Storage capacity from 250 kilobytes to 2.5 megabytes.
- Desk or rack mountable.
- Accomodates both single-sided and double-sided drives.
- Industrial quality cabinet with die cast front bezel.
- Drives pull out for easy service and maintenance.

Prices:

Cabinet with (2) single sided drives w/power supply . .	$1595.00
Cabinet with (2) double sided drives w/power supply .	$2295.00
Cabinet w/power supply	$ 495.00



Model V500

Vista's Line of High Performance, High Reliability Products also Includes these Advanced Components

Single Eight-Inch Drive Subsystem Model V500

Features:

- Up to 1.2 megabyte on-line storage capacity
- Compatible with Radio Shack Model II, Apple II® and most popular S100 based computer systems.
- Highly reliable power supply provides ±5% regulation and over-voltage protection.

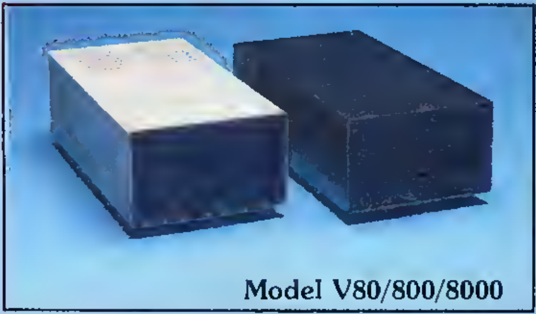
Prices: Starting as low as $900.00

Single Minifloppy Disk Drive Subsystem Model V80/800/8000

Features:

- From 102K to 816K bytes on a single drive.
- Fastest track-to-track access time on any minifloppy drive — 5 milliseconds.
- Dual-head technology on the V800/8000 allows on-line use of both sides of diskette

Prices: Starting as low as $395.00



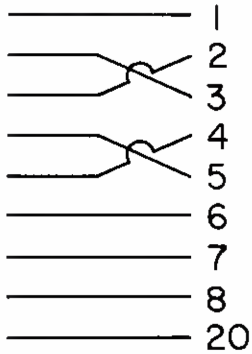
Model V80/800/8000

~65 **Vista Computer Company** 1317 E. Edinger Avenue • Santa Ana, CA. 92705 • (714) 953-0523

Dealer Inquiries Call Toll Free (800) 854-8017

"In spite of the delays and other minor problems, I still am satisfied with my Model I and Model II. . ."

	Printer Plug Pin Number	Model II Plug Pin Number
Ground	1	1 Ground
Transmitted Data	2	2 Transmitted Data
Received Data	3	3 Received Data
Request to Send	4	4 Request to Send
Clear to Send	5	5 Clear to Send
Data Set Ready	6	6 Data Set Ready
Signal Ground	7	7 Signal Ground
Carrier Detect	8	8 Carrier Detect
Terminal Ready	20	20 Terminal Ready



Instructions:

Remove the cover on the printer plug. The pin numbers are stamped inside the front part of the connector by the pins. Carefully note the pin numbers and the color of the wires connected to each pin.

Unsolder the wires from pins 2 and 3. Reverse the wires (82 to 3, 3, to 2) and resolder.

Unsolder the wires from pins 4 and 5. Reverse the wires (4 to 5, 5 to 4) and resolder.

Solder a small gauge wire from pin 6 to pin 8 to pin 20. Do not remove the wires already soldered to those pins.

Fig. 1. Pin Diagram of Xerox 1740 and TRS-80 Model II

across three of the pins. Fig. 1 shows the connections. The only problem I encountered in making the modifications was that the connector was of the crimp-on type. Consequently, I had to cut and splice the wires. If I were to do it again, I would replace the DB-25P connector with a solder tail type. It would even be possible to make a small jumper cable to install between the printer cable and the Model II. Fig. 2 shows such an adapter.

The printer is connected to Port B in the

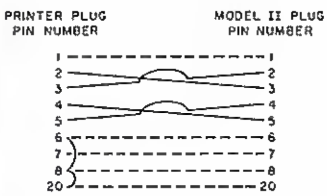


Fig. 2. Adapter cable from printer to Model II

Model II. It was selected because Port A is configured with the additional signals required for a modem. If you have a modem hooked to Port A, you can leave it there. If not, you will need a "dummy" plug in Port A. Fig. 3 shows such a shorting plug. This point has caused me some confusion. The confusion relates to the fact that some Model IIs seem to work without the dummy plug. So far, I have found at least three machines which work without the plug. I have discussed the problem with Radio Shack technical personnel. They were not aware of the fact, and as of yet have no explanation.

A minor complaint is that the serial port connectors do no have any type of devices to hold the plugs in place. The addition of a threaded post, such as the ones on the TI-810, would have cost another penny or two, but would make life easier. In fact, the connectors are attached to the plastic cabinet with small bolts and nuts which are not

"... you should not attempt these stunts unless you have your parents' approval and your unit is no longer under warrantee."

attached to the case. If you attempt to remove the connectors (as I did), the nuts fall inside the case and are murder to retrieve. Once you have loosened the bolts, you must open the case to tighten them.

Poor planning! The only solutions to the loose connector problem are epoxy, duct tape, or nerves of steel. If you bang the keys, the connector falls off. I settled for tape.

Problem Two: Software

Did I mention that the printer still didn't work with the Model II? Since I was absolutely sure that the hardware was correct, the problem had to be in the software. The first thing I discovered after resorting to reading the Model II manual was that TRSDOS 1.1 did not include serial port

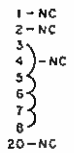


Fig. 3. Terminator plug for unused serial Port A.

driver software. I was faced with the chore of writing an assembly language driver. Fortunately, while I was still bracing myself for that chore, TRSDOS 1.2 arrived. It has serial port drivers. (Considering that I really needed TRSDOS 1.2 six months ago, Nestor's Law still stands.) Now all I had to do was learn how to use the serial port driver routine in TRSDOS 1.2.

Solution Two: Software

It wasn't really too difficult to learn to use the serial driver routine in TRSDOS 1.2. I learned by experiment that it would drive the Xerox 1740 reliably at speeds of 110, 300, and 600 baud. Since the printer operates at a maximum print speed of 45 characters per second, the 600 baud rate is the one I decided to use. I learned from the manual that the 1740 uses an eight-bit word, no parity, and one stop bit. The next experience was learning about the TRSDOS SETCOM command.

SETCOM is "SETs the COMmunications channels." It is issued from:

TRSDOS READY..

The correct command is as follows:

```
SETCOM B=(600,8,N,1), A=OFF
```

If you mis-typed the command, or for any other reason wish to change it, you must first turn the channels off. This is done with a SETCOM A=OFF,B=OFF command. If you use a reset, it is necessary to use the SETCOM sequence again. So much for SETCOM for the moment.

The next command to learn is FORMS. It sets the number of columns per line and lines per page. The defaults are for 132 columns per line and 60 lines printed out of the 66 lines per page. These are correct for 14 7/8" wide paper. The width would have to be reduced to 80 columns for 8 1/2" stock. A third command tells the Model II if your printer uses a special control code. I don't know what that really means, but so far I haven't found a printer that used it anyhow. The final option is between a parallel and serial printer. Parallel is the default, so we have to specify that this is a serial printer. If the FORMS command were entered in full it would read:

```
FORMS{P=66, W=132, L=60, C=0, S}
```

Since we are going to use the defaults for all but the last item, we can get by with:

```
FORMS {S}
```

The complete sequence is:

SETCOM B=(600,8,N,1),A=OFF	(SETCOM uses parentheses)
FORMS {S}	(FORMS uses lower case brackets)

If all is well, the display will respond with:

```
FORMS SET P=66 L=60 W=132 C=0
PRINTER READY? (Y/Q)... The answer is Y.
```

The printer should advance one line in response to a test signal sent from the Model II. If it doesn't and responds with ERROR 45, you have problems. ERROR 45 means that the printer is NOT READY. Check it out.

The next sequence is:

```
ALIGN PAPER TO TOP OF PAGE
PRESS ANY KEY TO CONTINUE
```

Align the paper and press a key. All is well with the world. The display will read:

```
Top, Repeat, or Quit?
```

This one really confused me. The correct answer it Q for Quit. It should return you to:

TRSDOS READY..

The serial port is now ready to drive the printer. You can use a DIR PRT command to print a disk directory. You can now go into BASIC and use LLIST or LPRINT at will. You can return to TRSDOS without disturbing the serial driver. However, if you reset the machine, the SETCOM and FORMS commands will have to be given again.

Solution Three: The DO File

Since it is cumbersome at best to type all of that stuff each time the machine is turned on, an easier way had to be found. The SETCOM and FORMS statements were incorporated into a DO file which I call PRINTER. The command BUILD PRINTER is used to create the file. SETCOM B=(600,8,N,1), A=OFF is entered as the first line. FORMS S is the second. Since I generally program in BASIC, I added a third line: BASIC MENU -F:5. It loads BASIC, allows for five files, and RUNs a program called MENU. You can modify the contents, but this is an example of a useful DO File. From TRSDOS you can accomplish all of these things by typing DO PRINTER.

If you want a true "turnkey" operation, type AUTO DO PRINTER. The machine will respond with AUTO SET 'DO PRINTER'. Whenever the Model II is turned on or reset, it will automatically setup the printer channel, load BASIC, and run the program.

Actually, the discussion of TRSDOS 1.2 is a bit off the track since there was not any sophisticated word processing software available to run under that operating system. However, I do use the printer with other programs which run under TRSDOS.

My shots at Radio Shack were taken in good fun. I realize that they can't be all things to all people. In spite of the delays and other minor problems, I still am satisfied with my Model I and Model II equipment. In fact, the personnel at the local store and repair center have been quite helpful. My thanks go to Dave Robinson, Dave Starkey, Lew Crawford, Joe Crossen, and Charles Brickenhauser for their efforts.

I suppose I should caution you that I am a professional computer hobbyist, and that you should not attempt these stunts unless you have your parents' approval and your unit is no longer under warrantee. If you are still interested, try it! ■

Achieve complete control of matrix print head needles through clever software manipulations.

A Tiger With Dots

George Somers
33 Deerfield Lane
Aberdeen, NJ 07747

A computer with only a CRT monitor is limiting—there are so many times when hard copy can be helpful, even essential. It wasn't long before I began to look for a printer. My final choice, after a great deal of comparison shopping, was the Integral Data IDS-440 impact Printer, known also as the Paper Tiger. The features included both a normal and enhanced mode for four different type densities. Yet, for me, the most attractive characteristic was an optional graphics feature that, if installed, would permit program control of each individual dot in the printhead matrix.

My first graphics programming project was to design a routine that would dump the contents of the TRS-80 video screen to the printer using the Tiger's dot control graphics feature to faithfully reproduce those unprintable graphic codes.

Software Control Codes

The IDS-440 achieves a great deal of its flexibility by employing a series of control codes: non-printable characters represented by 00H–1AH (0–31 decimal). Table 1 summarizes these software control codes. Control code 03H is very important because it allows the printer to suspend interpretation of all the subsequent bytes that it receives as printable ASCII characters, and enables the graphic function.

Raster Scan Dot Printing

Once in the graphics mode, the printer operates under a raster scan technique, printing columns of up to seven vertical dots across the page. Fig. 1 graphically documents this path across the page.

Individual Dot Access Via Software

The individual printhead needles (there are seven of them) are activated according to the printer's decomposition of the byte received into its component bits. If a 23H is received, then bits 0, 1, 2 and 4 are turned on and their respective printhead needles are energized to print dots. Due to the nature of the scan technique, the owner's manual recommends that bit 7 remain off (0) at all times. This is necessary because bit 0 of the

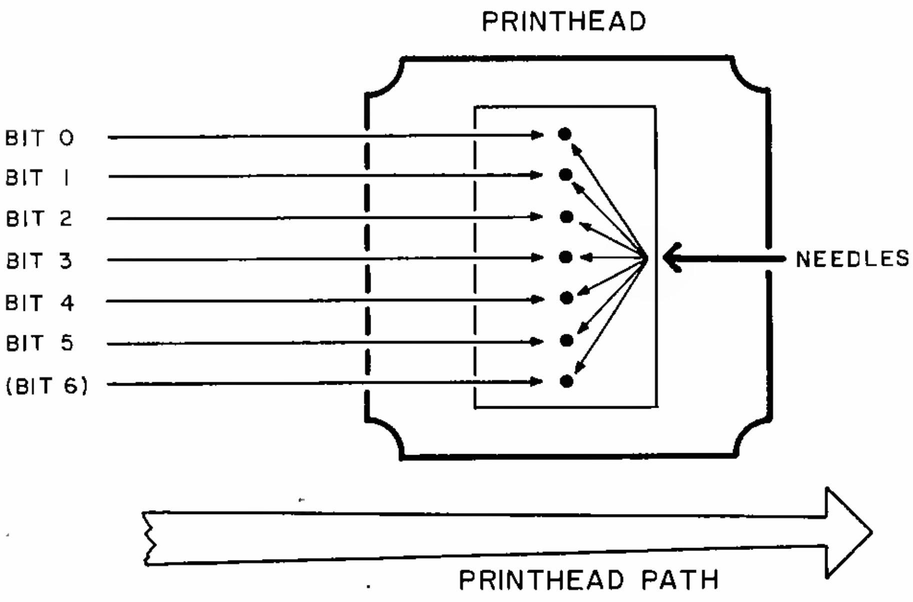


Fig. 1. Printhead Bit Pattern

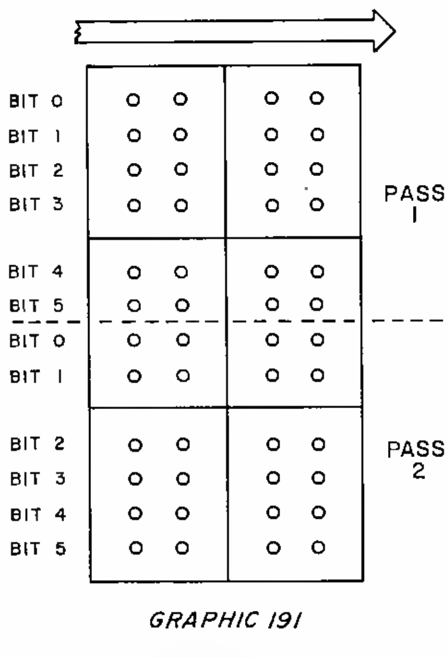


Fig. 2. Bits Superimposed on Graphic Block

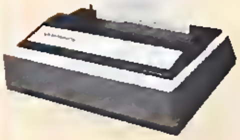
Orange Micro

"THE COMPUTER PRINTER SPECIALISTS"

UP TO 25% DISCOUNTS! — SAME DAY SHIPMENT!

CENTRONICS 737 (RADIO SHACK LINE PRINTER IV)

Word Processing Print Quality



• 18 x 9 dot matrix; suitable for word processing • Underlining • proportional spacing • right margin justification • serif typeface • 50/80 CPS • 9½" Pin Feed/Friction feed • Reverse Platen • 80/132 columns

CENTRONICS 737-1 (Parallel) ....(List $995) $765
CENTRONICS 737-3 (Serial) ....(List $1045) $815

VISTA — C. ITOH

Daisy Wheel Letter Quality

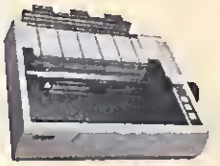
• 25 CPS (Optional 45 CPS) • Typewriter quality • Centronics parallel • RS 232 Serial (Optional) • Proportional spacing • Bidirectional • Programmable VFU • Self test • Diablo compatible • Friction feed (Optional tractors) • 136 printable columns. • Manufactured by C. ITOH.

VISTA V300 (C. ITOH) ....(List $1895) $ Call

EPSON MX80/MX70

OUR BEST BUY!

Low-Priced Professional Print Quality



• 9 x 9 dot matrix • Lower case descenders • 80 CPS • Bidirectional, Logic seeking • 40, 66, 80, 132 columns per line • 64 special graphic characters: TRS-80 Compatible • Forms handling • Multi-pass printing • Adjustable tractors

EPSON MX80 ....(List $645) $Call
EPSON MX 70 Dot graphics, 5 x 7 matrix......(List $450) $Call

ANACOM

Low Cost, High Speed, Wide Carriage

• 9 x 9 dot matrix • Lower case descenders • Wide carriage • Adjustable tractors to 16" • 150 CPS, Bidirectional, Logic Seeking

ANACOM 150 ....(List $1350) $ Call

ANADEX

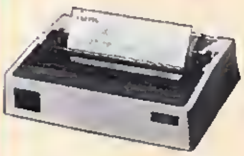
Dot Graphics, Wide Carriage

• 11 x 9 dot matrix; lower case descenders • Dot resolution graphics • Bi-directional, logic seeking • Up to 200 CPS • RS 232 Serial & Parallel • Forms control • X-ON/X-OFF • Up to 6 part copy.

ANADEX 9501...........................(List $1650) $1350

OKIDATA MICROLINE SERIES

TRS-80 Graphics Compatibility, Friction Feed



• 9 x 7 dot matrix • 80 CPS • 80, 132 columns — 64 shapes for charts, graphs & diagrams • Double wide characters • 6/8 lines per inch • Up to 3 part copy • Friction & pin feed • 200 M character head warranty

OKIDATA MICROLINE 80 .... (List $800) $520
OKIDATA M82 Bidirectional, Forms handling ....(List $960) $750
OKIDATA M83 Wide carriage, 9 x 9 dot matrix ..... (List $1260) $1050

NEC SPINWRITER

High Speed Letter Quality

• 55 CPS • Typewriter quality • Bidirectional • Plotting • proportional spacing.

5510-5 RO, Serial, w/tractors ....(List $2995) $2625
5530-5 RO, Parallel, w/tractors................(List $2970) $2599

TELEVIDEO CRT'S

AT DISCOUNT PRICES!

TVI 912C
TVI 920C
TVI 950
Please Call Toll Free — Prices are too low to advertise

QUANTITY PRICING AVAILABLE

IDS PAPER TIGERS

Dot Resolution Graphics, quality print, speed

• 7 wire printhead (445); 9 wire printhead (460) with lower case descenders • Over 150 CPS • bi-directional, logic seeking (460) • 8 character sizes; 80-132 columns • Adjustable tractors • High-resolution dot graphics • Proportional spacing & text justification (460).

IDS 445G 7 wire printhead, graphics ....(List $895) $ 750
IDS 460G 9 wire printhead, graphics ....(List $1394) $1150
IDS 560G 9 wire, wide carriage, graphics ......(List $1794) $1590

PRINTERS

MALIBU 165 wide carriage, graphics, letter quality .. (List $2495) $ 1975
QUME 5/45 typewriter quality ....(List $2905) $ 2559

INTERFACE EQUIPMENT

CCS APPLE PARALLEL Interface & cable .... $ 150
APPLE II - EPSON MX80 parallel interface board & cable .... $ 100
SSM AIO BOARD Apple Serial/parallel interface ...... (List $225) $ 175
MICROTRONICS Atari parallel interface....................... $ 69
ATARI 850 Interface module, serial/parallel .... $ 199
TRS-80 CABLES to keyboard or Exp. interface .... $ Call
NOVATION D-CAT direct connect modem.................... $ Call

CALL FOR FREE CATALOG

(800) 854-8275

CA, AK, HI (714) 630-3322

At Orange Micro, we try to fit the right printer to your application. Call our printer specialists for free consultation.

TELEPHONE ORDERS: Mon.-Fri. 8:30 - 5:00
The Orange Micro Printer Store (Retail):
Mon.-Fri. 10:00 - 6:00, Sat. til 4:00

Phone orders WELCOME; same day shipment. Free use of VISA & MASTERCARD. Personal checks require 2 weeks to clear. Manufacturer's warranty included on all equipment. Prices subject to revision.

Orange Micro, Inc.
3150 E. La Palma, Suite I
Anaheim, CA 92806

"My first awkward attempts at printing dots were far from successful. . . At seemingly random times, extraneous strings of dots were printed along with the legitimate ones."

ensuing printhead pass overlays bit 7 of the previous pass. No byte larger than 127 (equivalent to all six low order bits activated) or 7FH should be output to the printer.

Graphics Control Code Flag

Once in the graphics mode, control codes must be preceded by a 03H control code. This bars the printer from interpreting the intended control code as dot printing data. Escape from the graphics mode is accomplished by printing a 03H followed by either a 01H (enhanced mode control character) or a 02H (normal mode control code). The printer then resumes standard character printing.

Two Final Control Codes

Two final details conclude the preliminary programming instructions. The vertical tab code causes the printhead to terminate the present horizontal graphics scan and initiate the next scan six dots down on the page. Following the previously detailed conventions, the printer must receive both a 03H and a 0BH. Finally, to turn on bits 1 and 2 alone, two 03H bytes are required in order to signal the Paper Tiger's interpretive ROM that data 03H is intended and not a prefix to a control code.

On to the Programming

My first awkward attempts at printing dots were far from successful. It was soon apparent that, once in the graphics mode, the BASIC LPRINT command was useless as a means of sending graphic bytes to the Paper Tiger. At seemingly random times, extraneous strings of dots were printed along with the legitimate ones. The bit pattern of these extraneous dots corresponded to a 0AH—the standard control code for a line feed.

Obviously, upon receipt of a 0CH the ROM attempted to perform a form feed by outputting a string of line feeds, being unaware that the 0CH was not a control code in the graphics mode. POKEing the printer port (14312 decimal, 37E8H) with hundreds of random graphic bytes elicited no such unexpected results, confirming the diagnosis of incompatibility between the LPRINT command and the requirements of the printer while printing in the graphics mode.

How to Send the Graphic Bytes?

I then wrote a very short assembly language driver that would pass the byte to print from BASIC to the driver, check the ready status of the printer port, and, when all was in order, send the byte to the printer. After studying the TRS-80 ROM, I found that calling 05D1H would perform the printer port status check. A disassembled listing of this useful subroutine is in Program Listing 1. In addition, calling 0A7FH solved the problem of obtaining the byte to print (the byte is passed through the USR command variable to the HL register pair). The complete driver is in Program Listing 2.

Packing the Driver

For convenience and increased portability, I POKEd this short assembly language driver into a BASIC string. This saves loading a separate system tape every time graphics printing is desired. Graphics programs may be built around this string, or the string itself may be appended to a program that exists already. Program Listing 3 is a simple BASIC program that accomplishes the string packing.

Function	Hex	Decimal
Enhanced Mode Printing	01H	1
Normal Mode Printing	02H	2
Graphics Mode / Graphics Escape	03H	3
Line Feed	0AH	10
Vertical Tab	0BH	11
Form Feed	0CH	12
Carriage Return	0DH	13
Select Printer	11H	17
Deselect Printer	13H	19
8.3 Character per Inch	1CH	28
10 Character per Inch	1DH	29
12 Character per Inch	1EH	30
16.5 Character per Inch	1FH	31

Table 1. Software Control Codes

```
05D1    3AE837    LD     A,(37E8H)    ;GET PRINTER STATUS BYTE
05D4    E6F0      AND    0F0H         ;MASK FOR STATUS BIT
05D6    FE30      CP     30H          ;IS STATUS READY?
05D8    C9        RET                 ;RETURN TO COMPLETE CHECK
```

Program Listing 1. CALL 05D1H Disassembly

```
460FH    CD7F0A    CALL    0A7FH        ;PASS BYTE IN BASIC
                                         USR STATEMENT TO HL
4612     CDD105    CALL    051DH        ;CHECK PRINTER STATUS
4615     20FB      JR      NZ,4612H     ;IF NOT READY, GO CHECK
                                         STATUS AGAIN
4617     7D        LD      A,L          ;GET CHARACTER TO PRINT
                                         TO THE ACCUMULATOR
461B     32E837    LD      (37E8H),A    ;SEND BYTE TO PRINTER
461B     C9        RET                  ;NO VARIABLE TO RETURN
                                         TO BASIC; JUST RETURN
```

Program Listing 2. Mini-driver Assembly

```
99 ' ******************************************************************************
100 '       This short program will stuff the BASIC string PR$ with the short
101 ' assembly language driver to send the graphics bytes to the PAPER TIGER
102 '
103 ' After its job is done it deletes those lines that are not needed any longer
105 '******************************************************************************
110 '
50000 PR$= "                  " :' The assembly language mini-driver
50001 X = PEEK (VARPTR (PR$) + 2) * 256 + PEEK (VARPTR (PR$) + 1)
50002 FOR I = 1 TO 13 :' 13 bytes in the mini-driver
50003 READ J :' Get object code byte into J
50004 POKE X + I - 1, J :' POKE object code byte into string
50005 NEXT I :' Continue until all object code bytes are POKEd into PR$
50006 DELETE 50001 - 50007 :' Once string is POKEd these lines are not needed
50007 DATA 205, 127, 10, 205, 209, 5, 32, 251, 125, 50, 232, 55, 201 :' Object code
```

Program Listing 3. BASIC String Pack of Listing 2

"...converting graphic codes to a dot matrix representation of the blocks was the remaining programming roadblock."

Grephic Blocks to Dots

With the mechanics of sending the byte out of the way, determination of a practical algorithm for converting graphic codes to a dot matrix representation of the blocks was the remaining programming roadblock.

After studying the user's manual concerning the width and density of dots in the graphics mode, I estimated that a reasonable interpretation of the graphics block represented by decimal 191 (all six component pixels turned on) would consist of a dot matrix composed of twelve vertical dots by four horizontal dots in the 12 character per inch printing density. (See Fig. 2.) In this scheme, each pixel would be represented on paper as two horizontal dots by four vertical dots.

Using the POINT Command

In order to determine which individual pixels are set on the screen, the Microsoft BASIC ROM has provided the POINT command. It takes the form of POINT (X,Y), where X and Y are coordinates of the pixel to be examined for its on or off state. With the aid of this handy function, a program can determine the corresponding bits in the printhead matrix and turn them on twice (for two horizontal dots). The one complication is that the next lower pixel must also be checked to see if it is necessary to turn on the bottom two dots of the six-dot printhead pass as well.

A vertical tab must then be sent at the end of a printhead pass so that the next scan (to print the bottom half of the graphic block) can occur.

The Procedure Summarized

The analytical procedure can be summarized in this way:

- Examine each graphics block as three vertical and two horizontal pixels by using the POINT command.
- Print the dot representation in two scans of six vertical dots each, allowing four vertical dots and two horizontal dots for each individual pixel.

The First Program

Program Listing 4 is a BASIC language coding of the screen dump procedure described above. It is not intended to be run on its own, but, rather, to be added to those BASIC programs with graphics output that are worthy of transfer to hard copy. Access to the subroutine is done by a simple GOSUB50000 statement. To jump to this routine via a GOTO50000 statement would necessitate changing the Return statement in line number 50090 to an outright GOTO statement. However, logic flow always seems to remain clearer when minor routines are called through GOSUBs.

At the heart of the subroutine are two FOR—NEXT loops. The first loop (line

Program Listing 4. Graphics Driver (BASIC)

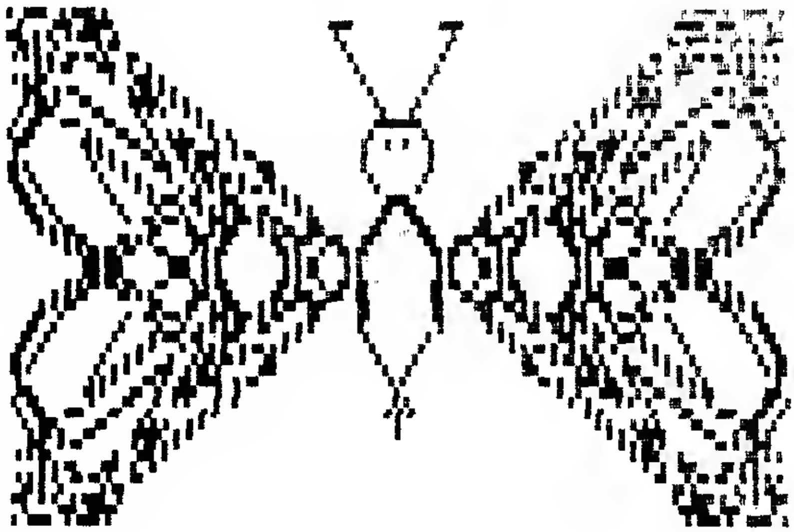
```
100 '**************************************************************************************
110 '                           BASIC PROGRAM # 1 - GRAPHICS PRINT DRIVER
120 '                         FOR THE INTEGRAL DATA IDS-440 - THE PAPER TIGER
130 '
140 ' ALL VARIABLES USED IN THIS ROUTINE BEGIN WITH THE LETTER "P"
150 ' THIS SUBROUTINE MUST BE CALLED BY AN APPROPRIATE GOSUB WITHIN THE MAIN PROGRAM
160 '**************************************************************************************
170 '
180 '
50000 PR$= "+
')2CVD7INKEY$ " :' The assembly language mini-driver
50001 PR% = VARPTR(PR$)+1: POKE16526,PEEK(PR%) : POKE16527,PEEK(PR%+1) :' POKE the USR address
50002 PU = USR (3) :' Enable PAPER TIGER's graphic mode
50003 PU = USR (3) : PU = USR (30) :' 12 Character per inch density
50004 '
50005 FOR PY = 0 TO 45 STEP 3 :' Horizontal line counter
50006    FOR PX = 0 TO 127 :' 1st pass vertical counter
50007          IF POINT (PX,PY) THEN PR = 15 :' If upper pixel
50008          IF POINT (PX,PY+1) THEN PR = PR + 48 :' If middle pixel
50010          PA$ = INKEY$ : IF PA$ <> "" THEN 50075 :' Stop printing on Keypress
50020          PU = USR (PR) : PU = USR (PR) :' Send graphic twice
50021          PR = 0 :' Re-initialize print character to 0
50022    NEXT PX :' Go back & continue until 1st scan is done
50023    PU = USR (3) : PU = USR (11) :' Do a vertical Tab
50025    FOR PX = 0 TO 127 :' 2nd pass Vertical counter
50030          IF POINT (PX,PY+1) THEN PR = 3 :' If middle pixel
50032          IF POINT (PX,PY+2) THEN PR = PR + 60 :' If bottom pixel
50040          IF PR = 3 THEN PU = USR (3) : PU = USR (3) :' Convert control 3 to data 3
50042          PU = USR (PR) : PU = USR (PR) :' Send graphic twice
50045          PR = 0 :' Re-initialize print character to 0
50047    NEXT PX :' Go back & continue until 2nd scan is done
50050    PU = USR (3) : PU = USR (11) :' Do Vertical Tab
50052 NEXT PY :' Continue until every horizontal line is done
50055 '
50075 PU = USR (3) : PU = USR (11) :' Do Vertical Tab
50078 PU = USR (3) : PU = USR (2) :' Convert to normal mode - escape graphics mode
50080 PU = USR (13) :' Do Carriage Return
50090 RETURN :' Return to the BASIC program location that called the print routine
```

Program Listing 5. Graphics and Character Driver (BASIC)

```
100 '**************************************************************************************
110 '                      BASIC PROGRAM # 2 - CHARACTER & GRAPHICS PRINT DRIVER
120 '                          FOR THE INTEGRAL DATA IDS-440 - THE PAPER TIGER
130 '
140 ' ALL VARIABLES USED IN THIS ROUTINE BEGIN WITH THE LETTER "P"
150 ' THIS SUBROUTINE MUST BE CALLED BY AN APPROPRIATE GOSUB WITHIN THE MAIN PROGRAM
151 '
152 'AUTO LINE FEED ON CARRIAGE RETURN FUNCTION MUST BE DEFEATED FOR THIS TO WORK CORRECTLY
153 'THAT IS DONE BY SETTING SWITCH 5 ON DIP SWITCH "S 4" TO OFF POSITION
154 '
155 'BE SURE AND CLEAR AN EXTRA 200 STRING SPACE FOR USE BY THIS ROUTINE
160 '**************************************************************************************
170 '
180 '
50000 PR$= "+
')2CVD7INKEY$ " :' The assembly language mini-driver
50001 PR% = VARPTR(PR$)+1: POKE16526,PEEK(PR%) : POKE16527,PEEK(PR%+1) :' POKE the USR address
50002 PU = USR (3) :' Enable PAPER TIGER's graphic mode
50003 PS = 15360 :' Initialize character print location
50100 '
50110 '
50120 FOR PY = 0 TO 45 STEP 3 :' Horizontal line counter
50121    PU = USR (3) : PU = USR (2) : PU = USR (30) :' Convert to normal mode
50122    PA$ = "" :' Initialize string that will hold characters to be printed
50123    FOR PB = PS TO PS + 63 :' Loop for 1 line of charcters
```

Program continues

```
50124       PA = PEEK (PB) : IF PA > 122 THEN PA = 32 :' Get char.; convert graphics to spaces
50125       PA$ = PA$ + CHR$ (PA) :' Concatenate new character to string to print
50126     NEXT PB :' Continue till every character on line is part of PA$
50127     LPRINT PA$ :' Print line of characters
50128     PS = PS + 64 :' PS now is start of next line of characters
50129     FU = USR (3) : FU = USR (3) : FU = USR (29) :' Convert back to graphics mode
50130     FOR PX = 0 TO 127 :' 1st pass vertical counter
50140         IF POINT (PX,PY) THEN PR = 15 :' If upper pixel
50150         IF POINT (PX,PY+1) THEN PR = PR + 48 :' If middle pixel
50160         PA$ = INKEY$ : IF PA$ <> "" THEN 50310 :' Stop printing on keypress
50161          The following two lines send a compensating graphic every 21 locations
50162         IF PX = 0 OR PX = 21 OR PX = 42 THEN PU = USR (PR) :' Compensate
50163         IF PX = 63 OR PX = 84 OR PX = 105 THEN PU = USR (PR) :' Compensate
50170         PU = USR (PR) : PU = USR (PR) :' Send graphic twice
50180         PR = 0 :' Re-initialize print character to 0
50190     NEXT PX :' Go back & continue until 1st scan is done
50200     PU = USR (3) : PU = USR (11) :' Do Vertical Tab
50210     FOR PX = 0 TO 127 :' 2nd pass Vertical counter
50220         IF POINT (PX,PY+1) THEN PR = 3 :' If middle pixel
50230         IF POINT (PX,PY+2) THEN PR = PR + 60 :' If bottom pixel
50240         IF PR = 3 THEN PU = USR (3) : PU = USR (3) :' Convert control 3 to data 3
50241         The next 2 lines compensate for disparity between char. & graphic size
50242       IF PX = 0 OR PX = 21 OR PX = 42 THEN PU = USR (PR) : IF PR = 3 THEN PU = USR (3)
50243       IF PX = 63 OR PX = 84 OR PX = 105 THEN PU = USR (PR) : IF PR = 3 THEN PU = USR (3)
50250         PU = USR (PR) : PU = USR (PR) :' Send graphic twice
50260         PR = 0 :' Re-initialize print character to 0
50270     NEXT PX :' Go back & continue until 2nd scan is done
50280     PU = USR (3) : PU = USR (11) :' Do Vertical Tab
50290 NEXT PY :' Continue until every horizontal line is done
50300 '
50310 PU = USR (3) : PU = USR (11) :' Do Vertical Tab
50320 PU = USR (3) : PU = USR (2) :' Convert to normal mode - escape graphics mode
50330 PU = USR (13) :' Do Carriage Return
50340 RETURN :' Return to the BASIC program location that called the print routine
```



Program Listing 6. Graphics Driver (Assembly)

```
                00010 ;THIS IS THE SOURCE CODE FOR A GRAPHICS SCREEN PRINTING
                00011 ;DRIVER FOR THE INTEGRAL DATA SYSTEMS IDS-440 PRINTER
                00012 ;            (NICKNAMED THE PAPER TIGER)
                00013 ;
                00014 ;THIS ASSEMBLY ROUTINE IS PRESENTLY LOCATED TO LOAD AT
                00015 ;THE TOP OF A 16K TRS-80. IT IS PROBABLY DESIRABLE FOR
                00016 ;THE USER TO CHANGE THE ORIGIN TO REFLECT THE TOP OF
                00017 ;HIS/HER MEMORY. CALL THIS PROGRAM VIA THE BASIC USR COMMAND.
                00018 ;
                00019 ;THE PROGRAM WILL EXAMINE EACH LOCATION OF THE TRS-80'S
                00020 ;VIDEO MONITOR AND WILL SEND TO THE PRINTER A DOT-MATRIX
                00021 ;REPRESENTATION OF THE GRAPHICS CODES (128 - 191).
                00022 ;CHARACTERS ARE NOT SUPPORTED BY THIS PROGRAM.
                00023 ;
                00024 ;IT IS SUGGESTED THAT THE DIP SWITCHES FOR PRINTING
                00025 ;CONTROL OF THE TIGER BE SET AS FOLLOWS:
                00026 ; 1-OFF  2-ON  3-OFF  4-OFF  5-ON  (DIP S 4)
                00027 ; 6-ON  (DIP S 3)
                00028 ;
7F40            00100         ORG     7F40H
7F40 3E03       00115 START   LD      A,3       ;GO INTO GRAPHIC MODE
7F42 CDC07F     00120         CALL    SEND
7F45 21013C     00125         LD      HL,3C01H        ;BECAUSE OF BASIC FORMAT
```

Program continues

```
7F48 2D        00130          DEC    L          ;NO 0 IN OBJECT ALLOWED
7F49 E5        00135 GOAGIN   PUSH   HL         ;SAVE LINE BEG. FOR 2ND PASS
7F4A AF        00140 GOAGAN   XOR    A          ;LD A WITH 0
7F4B CB7E      00145          BIT    7,(HL)     ;IS IT A GRAPHIC CODE?
7F4D 286D      00150          JR     Z,NOTONE          ;SEND 0'S IF NO
7F4F CB46      00155          BIT    0,(HL)     ;TOP LEFT BIT ON?
7F51 C4DB7F    00160          CALL   NZ,ADD15          ;LD A WITH 15 IF YES
7F54 CB56      00165          BIT    2,(HL)     ;MIDDLE LEFT BIT ON?
7F56 C4DB7F    00170          CALL   NZ,ADD48          ;LD A WITH 48 IF YES
7F59 CDC67F    00175          CALL   SENDER     ;BYTE TO PRINTER
7F5C AF        00180          XOR    A          ;LD A WITH 0
7F5D CB4E      00185          BIT    1,(HL)     ;TOP RIGHT BIT ON?
7F5F C4D87F    00190          CALL   NZ,ADD15          ;LD A WITH 15 IF YES
7F62 CB5E      00195          BIT    3,(HL)     ;MIDDLE RIGHT BIT ON?
7F64 C4DB7F    00200          CALL   NZ,ADD48          ;LD A WITH 48 IF YES
7F67 CDC67F    00205 NOTHIN   CALL   SENDER     ;BYTE TO PRINTER
7F6A 23        00210          INC    HL         ;NEXT SCREEN LOCATION
7F6B 7D        00215          LD     A,L        ;IS IT THE FIRST LOCATION OF
7F6C E63F      00220          AND    3FH        ;THE NEXT LINE?
7F6E 20DA      00225          JR     NZ,GOAGAN         ;GO BACK IF NOT
7F70 CDDE7F    00230          CALL   VERTAB     ;VERTICAL TAB AFTER 1ST PASS
7F73 E1        00235          POP    HL         ;SCREEN LOCATION FOR 2ND PASS
7F74 AF        00240 GONORE   XOR    A          ;LOAD A WITH 0
7F75 CB7E      00245          BIT    7,(HL)     ;IS IT A GRAPHIC CODE?
7F77 284B      00250          JR     Z,NOTGRA          ;SEND 0 IF NO
7F79 CB56      00255          BIT    2,(HL)     ;MIDDLE LEFT BIT ON?
7F7B C4E97F    00260          CALL   NZ,ADD3           ;LD A WITH 3 IF YES
7F7E CB66      00265          BIT    4,(HL)     ;BOTTOM LEFT BIT ON?
7F80 C4EC7F    00270          CALL   NZ,ADD60          ;LD A WITH 60 IF YES
7F83 FE03      00275          CP     3          ;IS IT TIGER CONTROL CODE?
7F85 CCEF7F    00280          CALL   Z,CTRLC    ;SEND AGAIN IF YES
7F88 CDC67F    00285          CALL   SENDER     ;BYTE TO PRINTER TWICE
7F8B AF        00290          XOR    A          ;LOAD A WITH 0
7F8C CB5E      00295          BIT    3,(HL)     ;MIDDLE RIGHT BIT ON?
7F8E C4E97F    00300          CALL   NZ,ADD3           ;LD A WITH 3 IF YES
7F91 CB6E      00305          BIT    5,(HL)     ;BOTTOM RIGHT BIT ON?
7F93 C4EC7F    00310          CALL   NZ,ADD60          ;LD A WITH 60 IF YES
7F96 FE03      00315          CP     3          ;IS IT TIGER CONTROL CODE?
7F98 CCEF7F    00320          CALL   Z,CTRLC    ;SEND AGAIN IF YES
7F9B CDC67F    00325 NOTGR    CALL   SENDER     ;BYTE TO PRINTER TWICE
7F9E 23        00330          INC    HL         ;NEXT SCREEN LOCATION
7F9F 7D        00335          LD     A,L        ;IS LOCATION 1ST ON NEXT LINE?
7FA0 E63F      00340          AND    3FH        ;          ""
7FA2 20D0      00345          JR     NZ,GONORE         ;GO BACK IF NOT
7FA4 CDDE7F    00350          CALL   VERTAB     ;VERTICAL TAB FOR 2ND PASS
7FA7 7C        00355          LD     A,H        ;END OF SCREEN
7FA8 FE40      00360          CP     40H        ;        MEMORY?
7FAA 209D      00365          JR     NZ,GOAGIN         ;GO BACK IF NO
7FAC 3E03      00370          LD     A,3        ;CONVERT
7FAE CDCD7F    00375          CALL   SEND       ;BACK
7FB1 3E02      00380          LD     A,2        ;TO
7FB3 CDCD7F    00385          CALL   SEND       ;NORMAL
7FB6 3E0D      00390          LD     A,0DH      ;SEND CARRIAGE
7FB8 CDCD7F    00395          CALL   SEND       ;        RETURN
7FBB C9        00410          RET
7FBC CDC67F    00415 NOTONE   CALL   SENDER     ;0 TO PRINTER TWICE
7FBF 18A6      00420          JR     NOTHIN     ;(1ST PASS NON-GRAPHIC)
7FC1 CDC67F    00425 NOTGRA   CALL   SENDER     ;0 TO PRINTER TWICE
7FC4 18D5      00430          JR     NOTGR      ;(2ND PASS NON-GRAPHIC)
7FC6 CDCD7F    00435 SENDER   CALL   SEND       ;ALWAYS SEND EACH
7FC9 CDCD7F    00440          CALL   SEND       ;        BYTE TWICE
7FCC C9        00445          RET
7FCD F5        00450 SEND     PUSH   AF         ;SAVE BYTE TO PRINT
7FCE CDD105    00455 SEND1    CALL   05D1H      ;PRINTER STATUS CHECK (ROM)
7FD1 20FB      00460          JR     NZ,SEND1          ;DON'T SEND IF NOT READY
7FD3 F1        00465          POP    AF         ;GET BACK BYTE TO PRINT
7FD4 32E837    00470          LD     (37E8H),A         ;SEND IT
7FD7 C9        00475          RET
7FD8 3E0F      00480 ADD15    LD     A,0FH      ;ADD 15 TO BYTE TO SEND
7FDA C9        00485          RET               ;(1ST PASS)
7FDB C630      00490 ADD48    ADD    A,30H      ;ADD 48 TO BYTE TO SEND
7FDD C9        00495          RET               ;(1ST PASS)
7FDE 3E03      00500 VERTAB   LD     A,3        ;DO
7FE0 CDCD7F    00505          CALL   SEND       ; A
7FE3 3E0B      00510          LD     A,0BH      ;  VERTICAL
7FE5 CDCD7F    00515          CALL   SEND       ;    TAB
7FE8 C9        00520          RET
7FE9 3E03      00525 ADD3     LD     A,3        ;ADD 3 TO BYTE TO SEND
7FEB C9        00530          RET               ;(2ND PASS)
7FEC C63C      00535 ADD60    ADD    A,3CH      ;ADD 60 TO BYTE TO SEND
7FEE C9        00540          RET               ;(2ND PASS)
7FEF CDC67F    00545 CTRLC    CALL   SENDER     ;TO CONVERT TO A DATA 3
7FF2 C9        00550          RET               ;SEND 3 TWICE
```


✓See List of Advertisers on page 306

"This kind of program would be... helpful for those programs whose numeric screen data are of equal importance to the graphics..."

50005) allows variable PY to hold the Y coordinate of the top pixel of each graphic block. It is incremented by steps of three in order to skip over the middle and bottom pixels of the graphic block, which are examined individually by the POINT statements within that loop. The second loop, which appears once for each of the two printhead passes (in lines 50006 and 50025), places the X coordinate of each screen pixel into variable PX.

PR$ (line 50000) holds the assembly language coding listed in Program Listing 2 after having been packed by the BASIC routine listed in Program Listing 3. The USR command pointer (locations 16526 and 16527 decimal) is POKEd with the address of this string-packed assembly program after the address is determined by the VARPTR function in line 50001. All subsequent USR calls are thus directed to the contents of PR$. The remainder of the program is self-explanatory with the aid of the various program comments.

Are Graphics Enough?

With a fully operative graphics program behind me, I attemped to revise the program in order to add supporting characters as well as graphics. This kind of program would be far more helpful for those programs whose numeric screen data are of equal importance to the graphics, as is the case with many business and scientific programs that employ screen graphs.

Two Problems to Solve

The addition of characters to the screen printing program brought with it two difficulties. The first solution was relatively easy. It necessitated defeating the automatic line feed upon receiving a carriage return—done by setting switch 5 on DIP switch S4 on the top of the Paper Tiger to its off position. This permits an initial pass for printing the screen characters, followed by a carriage return (now, minus the line feed), and then the required two passes for the graphics. Both were terminated by vertical tabs that, when combined, are equivalent to a line feed.

The second problem required a software solution and a necessary compromise in the final printed result. The problem was evident in the slight but significant incompatibility between the width of characters and the interpreted graphics blocks. It is a great deal less glaring when a compromise in the final printed screen is accepted; characters are printed in the 12 character per inch density and graphics are printed in the 10 character per inch density.

The printed graphics look a bit more faithful to the original screen graphics when they are printed at the 12 character per inch size, but for all practical purposes, this difference in appearance is insignificant, particularly in scientific and business graphic applications. Even with this adjustment, graphics can slightly overrun characters by the end of a printhead pass. I decided to compensate slightly every twenty-one horizontal pixels, for the minor disparity; again, with very little change in the final result.

The Second Program

Program Listing 5 is a listing of my results in the attempt to code a subroutine that would support both characters and graphics. Much of this program is the same as Program Listing 4. Lines 50121 through (and including) 50129 are the additions that handle the character printing. At the end of each character scan, PA$ holds the string of characters to be printed. Any non-characters encountered are converted into spaces (line 50124).

Because a string is being manipulated during this routine, it is wise to clear 200 extra bytes of string handling space beyond the main program's requirements. The compensation for the disparate printing of characters and graphics is accomplished both in lines 50163 and 50242–50243. In addition, since the compensation on the second pass (at line 50242) may result in a 03H, that

Program Listing 7. Graphics and Character Driver (Assembly)

```
                ;THIS IS THE SOURCE COOE FOR A SCREEN PRINTING ORIVER
                ;FOR THE INTEGRAL OATA SYSTENS IDS-440 PRINTER
                ;               (NICKNAMEO THE PAPER TIGER)
                ;
                ;THE PROGRAM WILL EXANINE EACH LOCATION OF THE TRS-BO'S
                ;VIDEO MONITOR ANO WILL SEND TO THE PRIHTER A DOT-MATRIX
                ;REPRESENTATION OF THE GRAPHICS COOES (128 - 191).
                ;IN ADOITIOH, IT SUPPORTS ALL CHARACTERS THAT ARE PRINTABLE
                ;TO THE TRS SCREEN.
                ;
                ;THE ROUTINE PRESEHTLY RESIDES AT THE TOP OF MEMORY FOR A
                ;16K TRS-BO. A USER WITH MORE MEMORY SHOULD CHANGE THE ORIGIN
                ;TO REFLECT THAT HAPPY FACT.
                ;
                ;IT IS SUGGESTEO THAT THE OIP SWITCHES FOR PRINTING
                ;CONTROL OF THE TIGER BE SET AS FOLLOWS:
                ; 1-ON  2-OFF  3-OFF  4-OFF  5-OFF  (OIP S 4)
                ; 6-ON  (OIP S 3)
                ;
7F00                          ORG     7F00H
7F00 COF07F                   CALL    LOADA3  ;GO INTO GRAPHICS NOOE
7F03 21013C                   LO      HL,3C01H        ;BECAUSE OF 8ASIC FORMAT
7F06 20                       OEC     L       ;NO 0 IN OBJECT ALLOWED
7F07 E5               GOAGIN  PUSH    HL      ;SAVE LINE BEG. FOR 2NO PASS
7F0B COBF7F                   CALL    INIT    ;INITIALIZE CHARACTER PRINTING
7F0B 3E1E                     LO      A,1EH   ;   12 CPI
7F00 COAB7F                   CALL    SENO    ;   FOR CHARACTER
7F10 7E               NEXTCH  LO      A,(HL)  ;GET CHAR. TO A
7F11 CB7F                     BIT     7,A     ;IS IT GRAPHIC?
7F13 2802                     JR      Z,OUTPUT        ;IF NO, JUMP
7F15 3E20                     LO      A,20H   ;IF YES, PRINT A BLANK
7F17 COAB7F           OUTPUT  CALL    SENO    ;SENO ONLY ONCE
7F1A 23                       INC     HL      ;NEXT SCREEN LOCATION
7F1B 70                       LO      A,L     ;   IS IT THE
7F1C E63F                     ANO     3FH     ;   THE ENO OF THE LINE?
7F1E 20F0                     JR      NZ,NEXTCH       ;IF NO, GO BACK & CONT.
7F20 COC77F                   CALL    INIT1   ;SENO CARRIAGE RETURN
7F23 3E03                     LO      A,3     ;CONVERT TO
7F25 COA47F                   CALL    SENOER  ;       GRAPHICS SIZE
7F28 3E10                     LO      A,10H   ;       10 CPI
```

Program continues

✓See List of Advertisers on page 306

". . . these programs do their respective jobs efficiently, but no program should ever pretend to be the final word on a computing procedure."

```
7F2A CDAB7F   00490         CALL  SEND   ;       ..
7F2D E1       00500         POP   HL     ;GET BACK LINE START FOR 1ST PASS
7F2E E5       00510         PUSH  HL     ;SAVE IT FOR 2ND PASS OF GRAPHICS
7F2F 060A     00520         LD    B,0AH  ;INITIALIZE GRA. COMP. COUNTER
7F31 AF       00530 GOAGAN  XOR   A      ;LD A WITH 0
7F32 CB7E     00540         BIT   7,(HL) ;IS IT A GRAPHIC CODE?
7F34 2864     00550         JR    Z,NOTONE      ;SEND 0'S IF NO
7F36 CB46     00560         BIT   0,(HL) ;TOP LEFT BIT ON?
7F38 2802     00570         JR    Z,NOTL ;IF NO, JUMP
7F3A 3E0F     00580         LD    A,0FH  ;IF YES, ADD 15 TO BYTE TO SEND
7F3C CB56     00590 NOTL    BIT   2,(HL) ;MIDDLE LEFT BIT ON?
7F3E 2802     00600         JR    Z,NOML ;IF NO, JUMP
7F40 C630     00610         ADD   A,30H  ;IF YES, ADD 48 TO BYTE TO SEND
7F42 CDA47F   00620 NOML    CALL  SENDER ;BYTE TO PRINTER
7F45 AF       00630         XOR   A      ;LD A WITH 0
7F46 CB4E     00640         BIT   1,(HL) ;TOP RIGHT BIT ON?
7F48 2802     00650         JR    Z,NOTR ;IF NO, JUMP
7F4A 3E0F     00660         LD    A,0FH  ;IF YES, ADD 15 TO BYTE TO SEND
7F4C CB5E     00670 NOTR    BIT   3,(HL) ;MIDDLE RIGHT BIT ON?
7F4E 2802     00680         JR    Z,NOTHIN      ;IF NO, JUMP
7F50 C630     00690         ADD   A,30H  ;IF YES, ADD 48 TO BYTE TO SEND
7F52 CDE57F   00700 NOTHIN  CALL  SENCOM ;PRINT, COMP., EOL CHECK
7F55 20DA     00710         JR    NZ,GOAGAN     ;GO BACK IF NOT
7F57 CDB67F   00720         CALL  VERTAB ;VERTICAL TAB AFTER 1ST PASS
7F5A E1       00730         POP   HL     ;SCREEN LOCATION FOR 2ND PASS
7F5B 060A     00740         LD    B,0AH  ;INITIALIZE GRA. COMP. COUNTER
7F5D AF       00750 GOMORE  XOR   A      ;LOAD A WITH 0
7F5E CB7E     00760         BIT   7,(HL) ;IS IT A GRAPHIC CODE?
7F60 2830     00770         JR    Z,NOTGRA      ;SEND 0 IF NO
7F62 CB56     00780         BIT   2,(HL) ;MIDDLE LEFT BIT ON?
7F64 2802     00790         JR    Z,NOMLL ;IF NO, JUMP
7F66 3E03     00800         LD    A,3    ;IF YES, ADD 3 TO BYTE TO SEND
7F68 CB66     00810 NOMLL   BIT   4,(HL) ;BOTTOM LEFT BIT ON?
7F6A 2802     00820         JR    Z,NOBLL ;IF NO, JUMP
7F6C C63C     00830         ADD   A,3CH  ;IF YES, ADD 60 TO BYTE TO SEND
7F6E FE03     00840 NOBLL   CP    3      ;IS IT TIGER CONTROL CODE?
7F70 CCA47F   00850         CALL  Z,SENDER      ;SEND AGAIN IF YES
7F73 CDA47F   00860         CALL  SENDER ;BYTE TO PRINTER TWICE
7F76 AF       00870         XOR   A      ;LOAD A WITH 0
7F77 CB5E     00880         BIT   3,(HL) ;MIDDLE RIGHT BIT ON?
7F79 2802     00890         JR    Z,NOMRB ;IF NO, JUMP
7F7B 3E03     00900         LD    A,3    ;IF YES, ADD 3 TO BYTE TO SEND
7F7D CB6E     00910 NOMRB   BIT   5,(HL) ;BOTTOM RIGHT BIT ON?
7F7F 2802     00920         JR    Z,NOBR ;IF NO, JUMP
7F81 C63C     00930         ADD   A,3CH  ;IF YES, ADD 60 TO BYTE TO SEND
7F83 FE03     00940 NOBR    CP    3      ;IS IT TIGER CONTROL CODE?
7F85 CCA47F   00950         CALL  Z,SENDER      ;SEND AGAIN IF YES
7F88 CDE57F   00960 NOTGR   CALL  SENCON ;PRINT, COMP., EOL CHECK
7F8B 20D0     00970         JR    NZ,GOMORE     ;GO BACK IF NOT
7F8D CDB67F   00980         CALL  VERTAB ;VERTICAL TAB FOR 2ND PASS
7F90 7C       00990         LD    A,H    ;END OF SCREEN
7F91 FE40     01000         CP    40H    ;       MEMORY?
7F93 C2077F   01010         JP    NZ,GOAGIN     ;GO BACK IF NO
7F96 CDBF7F   01020         CALL  INIT   ;CONVERT TO NORMAL, SEND CAR RET
7F99 C9       01050         RET
7F9A CDA47F   01060 NOTONE  CALL  SENDER ;0 TO PRINTER TWICE
7F9D 18B3     01070         JR    NOTHIN ;(1ST PASS NON-GRAPHIC)
7F9F CDA47F   01080 NOTGRA  CALL  SENDER ;0 TO PRINTER TWICE
7FA2 18E4     01090         JR    NOTGR  ;(2ND PASS NON-GRAPHIC)
7FA4 CDAB7F   01100 SENDER  CALL  SEND   ;ALWAYS SEND EACH
7FA7 CDAB7F   01110         CALL  SEND   ;       BYTE TWICE
7FAA C9       01120         RET
7FAB F5       01130 SEND    PUSH  AF     ;SAVE BYTE TO PRINT
7FAC CD0105   01140 SEND1   CALL  05D1H  ;PRINTER STATUS CHECK (ROM)
7FAF 20FB     01150         JR    NZ,SEND1      ;DON'T SEND IF NOT READY
7FB1 F1       01160         POP   AF     ;GET BACK BYTE TO PRINT
7FB2 32E837   01170         LD    (37E8H),A     ;SEND IT
```

Program continues

situation is also provided for on the same lines. All other program operations are documented in the comment statements.

These programs require that switch 6 of DIP switch S3 be set to its on position in order to allow the Paper Tiger to receive the software control codes.

The Speed Factor

As both of these programs are written in BASIC, they tend to run slowly, particularly Program Listing 5. Because of this, I soon translated both programs into assembly language code, and include my results here as Program Listing 6 (graphics only) and Program Listing 7 (both characters and graphics) for those readers who wish to maximize the speed aspect of the printer. It is not within the scope of this article to discuss details of the assembly language translations of the BASIC programs; the assembly techniques employed closely parallel those of their slower counterparts, and the listings are commented liberally to assist in their deciphering by willing readers. Comparisons of run time speed for one full screen dump will all pixels turned on follow:

Graphics only:
- Program Listing 4 (BASIC) : 4 ½ minutes
- Program Listing 5 (Assembly) : 25 seconds

Graphics and characters:
- Program Listing 5 (BASIC) : 8 1/2 minutes
- Program Listing 6 (Assembly) : 48 seconds

Parting Comments

All of these programs do their respective jobs efficiently, but no program should ever pretend to be the final word on a computing procedure. There can always be a better way. With this in mind, I encourage readers to experiment, make improvements, and add features. For instance, with very little effort, both of the assembly language programs could be shortened (though, perhaps, suffering a loss of clarity in the process) and stuffed into a BASIC string so that they may be as portable as the little driver already present in the two BASIC programs.

I will be very happy to respond to any comments, questions, and suggestions that are addressed to me. And now, it's back to work. You see, I have visualized a custom designed character set, and since I can control those dots. . . ! ■

```
7F85 C9                    RET
7FB6 CDF07F       VERTA8   CALL    LOADA3   ;DO A
7FB9 3E0B                  LD      A,0BH    ;  VERTICAL
7FBB CDAB7F                CALL    SEND     ;   TAB
7FBE C9                    RET
7FBF CDF07F       INIT     CALL    LOADA3   ;CONVERT TO
7FC2 3E02                  LD      A,2      ;           NORMAL
7FC4 CDAB7F                CALL    SEND     ;         ""
7FC7 3E0D         INIT1    LD      A,0DH    ;SEND
7FC9 CDAB7F                CALL    SEND     ;    CARRIAGE
7FCC C9                    RET              ;             RETURN
7FCD F5           COMPEN   PUSH    AF       ;SAVE GRAPHIC TO PRINT
7FCE 78                    LD      A,B      ;GET COMPENSATION COUNTER(8) TO A
7FCF FE0A                  CP      0AH      ;EVERY 10 SEND COMPENS. GRAPHIC
7FD1 200F                  JR      NZ,LEAVE         ;IF NOT LEAVE
7FD3 F1                    POP     AF       ;GET GRAPHIC BACK
7FD4 0601                  LD      B,1      ;CLEAR THE COUNTER
7FD6 05                    DEC     B        ;(0 NO GOOD IN BASIC FORMAT)
7FD7 FE03                  CP      3        ;IS IT CONTROL-C?
7FD9 2003                  JR      NZ,NOTCON        ;IF NOT SEND ONLY ONCE
7FDB CDAB7F                CALL    SEND     ;SEND HERE ONLY ON CONTROL-C
7FDE CDAB7F       NOTCON   CALL    SEND     ;SEND COMPENSATING GRAPHIC
7FE1 C9                    RET              ;IF COMPENSATED RETURN
7FE2 04           LEAVE    INC     B        ;INCREMENT COMPENS. COUNTER
7FE3 F1                    POP     AF       ;GET PRINTING GRAPHIC BACK
7FE4 C9                    RET              ;RETURN ON NO COMPENSATION
7FE5 CDA47F       SENCOM   CALL    SENDER   ;BYTE TO PRINTER TWICE
7FEB CDCD7F                CALL    COMPEN   ;CHECK FOR COMPENSATION
7FEB 23                    INC     HL       ;NEXT SCREEN LOCATION
7FEC 7D                    LD      A,L      ;IS LOCATION FIRST
7FED E63F                  AND     3FH      ;               ON NEXT LINE?
7FEF C9                    RET              ;GO BACK WITH Z-YES, NZ-NO
7FF0 3E03         LOADA3   LD      A,3      ;CONTROL-C TO A
7FF2 CDAB7F                CALL    SEND     ;SEND TO PRINTER
7FF5 C9                    RET
0000                       END
00000 TOTAL ERRORS
```


✓See List of Advertisers on page 306

BUSINESS SOFTWARE

Program	Media	Machine	Price
Lazy Writer	D	Mod I	$125.00
Lazy Writer.	D	Mod III	$175.00
Postman Data Handler	D	Mod I	$125.00
Graphix	D	Mod I or III	$49.95
Critpath	D	Mod I	$600.00
Critpath	D	Mod III	$750.00
Compleat Idiots Bookeeper	D	Mod I or III	$49.95

UTILITY SOFTWARE

Program	Media	Machine	Price
Super Utility	D	Mod I	$49.95
Super Utility II	D	Mod I	Call
Quick Fix	D	Mod I	$34.95
Chain Maker (for ND-80 only)	D	Mod I	$14.95
Tape Copy 1	D	PMC/Mod I	$14.95
Tape Copy 2	D	Mod I and III	$14.95
Hyres	D	Mod I	$9.95
Bug +	D	Mod I and III	$14.95
Make 80	D	Mod I	$14.95
Double Zap II ND-80	D	Mod I	$49.95
Double Zap II LDOS	D	Mod I	$39.95
Easy Term	D	Mod I or III	$59.95

GAMES

Program	Media	Machine	Price
Space Colony	C	Mod I	$14.95
TRS Super Invaders	C	Mod I	$19.95
Super Vaders	C/D	Mod III	$19.95
Attack Force	C	Mod I or III	$15.95
Galaxy Invasion	C	Mod I or III	$15.95
Super Nova	C	Mod I or III	$15.95
Meteor Mission II	C	Mod I or III	$15.59
Cosmic Fighter	C	Mod I or III	$15.95
Symon Capture	C	Mod I or III	$12.95
Flippy 1.3	C	Mod I or III	$14.95

All the game prices are for the cassette version only. The disk version is extra. Call for price.

OPERATING SYSTEMS

Program	Media	Machine	Price
NEWDOS/80	D	Mod I	$149.95
LDOS	D	Mod I	$139.95

BOOKS

Program	Media	Machine	Price
Mod III Rom Commented	Book	Mod III	$22.50 + 2.50 Shipping

SPECIAL

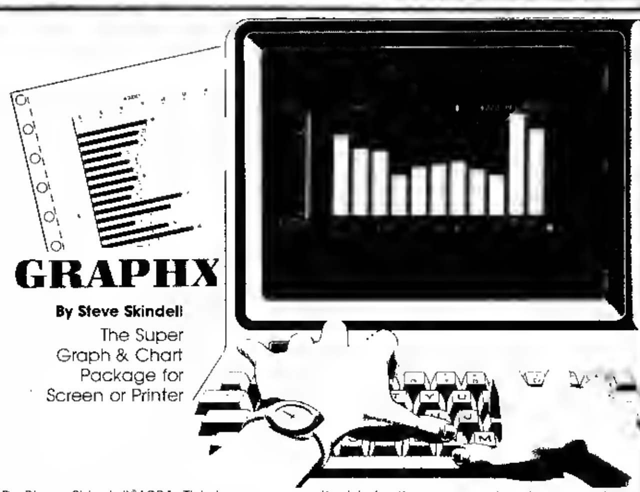
If you purchase NEWDOS/80 receive CHAIN MAKER at now extra charge!
Only $149.00
plus $2.50 shipping and handling

Purchase LAZY WRITER & POSTMAN DATA HANDLER (Mod I version only)
for $250.00
RECEIVE POSTWRITER AT NO EXTRA CHARGE
This month only

SUPER SPECIAL OF THE MONTH

Purchase any business package from Soft Sector Marketing, Inc. this month and receive QUICK FIX for only **$25.00**
This is a $34.95 value.

MasterCard **SEND FOR OUR NEW FREE CATALOG!** VISA
Or catalog is included with all orders.
See last month's 80 Micro for more details on our software or send for overviews.



GRAPHX
By Steve Skindell
The Super Graph & Chart Package for Screen or Printer

By Steve Skindell©1981. This is a program that is for the person who does reports or requires some sort of plotted output to show gains or losses, or any type of output that needs graphs. This program puts to the screen or to a printer the plotted points in bar graph format and plots the mean, and the averages of input. This is a very important program for accountants, CPS's and the average businessman to evaluate, at a moments glance where he is, was, or where he is going. Files saved to disk can be recalled at any time to be reexamined, modified, or just reprinted. An extra feature: if you have the Microline 80 printer, by Okidata or Epson MX-80, your output is in true graphics. Information is supplied for the user so he can modify this program for other printers. Comes complete and ready to run. Requires MOD I or MOD III, 48K disk. Printer optional (132 col.). Only **$49.95**

MX-80 PACKAGE SPECIAL!

MX-80* – New
You Pay Only **$599.00**
Save Up To **$162.95***
Hurry – Supply Limited!

And At No Extra Cost To You:
1 Box of Paper (2700 sheets 20# white) - 2 Extra Ribbons - 1 Extra Print Head and Shipping Included for all UPS in U.S.
***Compared to all items at list price!**
*for MX-80 f/t add $150.00
Sorry, no C.O.D. on this special. To receive this discount you must prepay
If you don't require all these extra items, call for price.

HOW TO ORDER

SSM SOFT SECTOR MARKETING, INCORPORATED
6250 Middlebelt • Garden City, MI 48135 • 1 (313) 425-4020

Dealer Inquiries Invited. ✓495

C.O.D. - certified check, M.O. or cash only. Sorry, no C.O.D. over $150.00! Most orders shipped next day. All orders must have shipping included. Please add 2% or $2.50, whichever is higher, for shipping (unless otherwise stated). Out-of-country orders, please add $10.00 additional shipping and handling, balance returned upon shipment.

*TRS-80 is a product of Radio Shack, division of the Tandy Corporation

A review of those cryptic error messages your 80 keeps sending you.

To Err Is. . . Forbidden

John D. Adams
13126 Tripoli Ave.
Sylmar, CA 91342

Easy as it is to use, the TRS-80 is an exacting little tyrant when it comes to following the rules. If I wrote here that this article was prepared on a *tiperiter*, you might lift your eyebrows a little, but from the context you'd know what I meant. Although the human mind may not be bothered by little details such as this, the computer is. Therefore, the computer needs ways of dealing with human error and omission.

The '80 has as complete a set of error handling routines, as might be expected from a computer in its price range. From the old WHAT, HOW and SORRY of Level I we jump to a set of 23 messages that are far more specific. Let's examine these messages in detail.

Error Codes

NF (error code one)—NEXT without FOR: We can get a lot of work done efficiently by using a loop that automatically increments or decrements. Consider the following lines:

```
10 FOR X=1 TO 100
20 ..........
30 NEXT
```

When line 10 is executed, the value 1 will be set in variable location X, and the value 100 will be stored internally. Information must also be stored to point to line 10, so the computer will know where to go for the next pass of the loop when line 30 has been executed. If the computer reads a NEXT instruction and no line contains FOR to store that pointer, it will not know where to go, and the NF message is returned.

Using FOR without NEXT will not return this error message, but neither will the loop cycle. At times the TRS-80 will return an NF error when loop structure is correct: This is an idiosyncrasy of the machine. (See Hubert Borrman's article "The 'Next' Trap" in the September 1980 *80 Microcomputing*.)

SN (error code two)—Syntax Error: This is probably the most frequently encountered error message. Syntax refers to the terms and symbols of a language, what they mean, and how they are arranged to produce logical results.

The part of your computer that translates BASIC into machine language is simpleminded; if we deviate ever so slightly from the rules it will not compensate. Common causes of syntax errors are:

Misspelled Instructions: Entering PRIBT instead of PRINT.

Misuse of Delimiters: Symbols such as quotation marks, semicolons, colons, and commas mean more than punctuation to the computer.

Illegal Signs: Trying to run the line, 10A = 3 x 4, will not get you the product you want. For example, an asterisk is used to indicate multiplication in BASIC. Such signs as the slash bar and up arrow have specific operational meanings, and no substitutes are accepted.

Illegal Statements: Dialects of BASIC differ, some contain words that others don't. Using terms such as CALL or PLOT in Level II BASIC will generate a syntax error.

Unmatched Parentheses: We often use multiple levels of parentheses to indicate how we want expressions evaluated. These symbols must be matched: one close parenthesis for each open parenthesis.

Improper Notation of Arguments: Many instructions in BASIC require some information be given afterward. This information is called the argument. Some arguments, such as AUTO mm,nn and DEFSTR n, do not require parentheses, while others, such as MID$(n$,p,n), SET(x,y) and SQR(x), do. Punctuation is critical in some arguments; for the PRINT@ n instruction, a comma must follow the display position. Appendix A of your manual sets forth the correct notation for all arguments.

Faulty Program Line Construction: Computer evaluation of mathematical expressions is done on the right side of the equal sign, and results are assigned to the left, so equations must be formatted accordingly. The line, 100 PRINT USING A$ TAB(30);B, is an invalid expression, but 100 PRINT TAB (30) USING A$;B is acceptable. Writing comprehensive guidelines for line construction is a difficult task—experience is the better teacher. If a line will not work as written, experiment with other arrangements.

When the computer finds a syntax error it automatically goes into edit mode and supplies the line number in which the error occurred. Typing an L will furnish the com-

"The part of your computer that translates BASIC into machine language is simple minded. . . ."

plete line. All other error messages shift the computer into command mode, returning the prompt sign and ready message.

Stored Locations

RG (error code three)—RETURN without GOSUB: This is a paired command, similar to the FOR-NEXT duo. Subroutines are sometimes placed at the end of the main program. When GOSUB is executed, its location is stored so the computer can return to the right place after the subroutine's termination. If a RETURN is read and there is no GOSUB location stored, the program can't continue, as it has no re-entry point. This can happen when we revise a program and delete or change GOSUB lines and forget about the RETURN.

OD (error code four)—Out of Data: READ-DATA instructions are used when many constants must be stored in the body of the program to be read when needed. If there are fewer items in the data line than called for in the READ line, this error message is returned. An example would be:

```
10 READ A,B,C
20 DATA 5,8
```

When the computer attempts to load location C, there is no data available. This error can also occur when an INPUT #-n instruction calls for data not available from tape. The OD message should not be confused with the double question mark (??), which indicates insufficient response to an INPUT instruction. In this case more data is needed.

FC (error code five)—Illegal Function Call: This indicates a command which is mathematically impossible or not within the capability of the machine or the language. For example: requesting the square root or logarithm of a negative number, using a negative number in the argument of the LEFT$, RIGHT$ or MID$ instructions, or setting up an array with negative dimensions. Many of us use hybrid programs in which the USR instruction is used to access a routine in machine language from BASIC. If the entry point of the machine language routine is not specified, an FC message is returned.

OV (error code six)—Overflow: There are limits to the value of numbers the TRS-80 can process. Integers must be in the range -32768 and $+32767$ inclusive. Single and double precision numbers must be in the range -1.7×10^{38} and $+1.7 \times 10^{38}$ inclusive. The latter are given in exponential notation which is a sort of shorthand way of writing very large, or very small, numbers. To convert, move the decimal to the right—or to the left, if the exponent of 10 is negative—the number of places designated by the exponent of 10. Thus, 1.7×10^{38} becomes 17 followed by 37 zeros. Fortunately, this range is sufficient for most programs.

OM (error code seven)—Out of Memory: This message indicates that all of the available RAM has been used up or reserved. This can come as a surprise if you are not aware of what is called "memory overhead." Each time we write a program line, five bytes are used, in addition to the byte space used to store the characters in the line: two for the line number, two for the line pointer and one for the carriage return. Dimensioning arrays, specifying double precision, using string variables, FOR-NEXT loops, parentheses and GOSUBs all eat up RAM. Programs which are running short on RAM space can be streamlined—see "Saving Memory Space" on page 11/1 of your manual.

UL (error code eight)—Undefined line: Instructions have been given to branch to a line which does not exist. This happens in program revision when line numbers have been changed. Also, make sure all branch instructions have valid destination points.

BS (error code nine)—Beyond Subscript: The individual spaces in an array are designated by using subscript numbers which are included in parentheses following the variable name. The location A$(12) indicates the 12th element or member in the string array called A$.

On power-up, the TRS-80 is prepared to handle the 11 element array. Larger arrays must be dimensioned using the DIM instruction. If an array has been dimensioned at 100, it will accept only that many elements. Attempts to enter additional elements return the BS message. Arrays should be properly dimensioned at or near the beginning of the program.

DD (error code 10)—Redimensioned Array: The process of setting up an array is complex, and once done, may not be redimensioned. If there is no way to avoid a size change, set up a new array with correct specifications and transfer the contents from the old to the new by using the matrix transposition routines found on page 6/5 of the manual.

Division by Zero

/0 (error code 11)—Division by Zero: The laws of mathematics state that division by zero is undefined. This error occurs when division is being performed with numbers generated in another part of the program. A screening line such as, 200 IF A = 0 THEN GOTO . . . may be used to bypass this situation.

ID (error code 12)—Illegal Direct: The INPUT instruction is for use in the execute mode of your computer, and may not be used as a direct statement. There are other ways of getting data entered: It may be embedded in the program (10 PI = 3.1416), or it may be read (10 READ PI:DATA 3.1416). INPUT stops execution and can be used with manual keyboard entry only while the program is running.

TM (error code 13)—Type Mismatch: An attempt has been made to use string data in a non-string application or vice versa. In the line 10 A = JANUARY, the variable A is for non-string data (value data); to load a word it should be named A$. Loading a number into a string location and then trying to use that number for computation will also generate this message.

OS (error code 14)—Out of String Space: Check the memory map of the 16K Level II TRS-80 on page D/2 of your manual. String data is stored from the top of memory down, and below that, the stack is built downwards. It is necessary to set some boundary between the two. On power-up you are allotted 50 bytes for string storage. The OS message is returned if that allotment is exceeded. If more string space is needed, reserve it by using the CLEAR n instruction at the beginning of the program. Be careful: If you execute this instruction after data has been entered, that data will be lost.

LS (error code 15)—String too long: You have exceeded the maximum string length of 255 characters. Break the string up into smaller units.

ST (error code 16)—String Formula Too Complex: If the computer decides the operation you are requesting is beyond the capability of the microprocessor, it will return this message. Rewrite the operation so it is handled in simpler blocks.

CN (error code 17)—Cannot Continue: When it is necessary or desirable to stop program execution, the CONT instruction allows us to continue program flow without loss of stored data. It may *not* be used after the edit mode has been entered or lines have been added or deleted. An error trapping routine is the best solution for the problem. Programs cannot be continued after termination with the END instruction.

NR (error code 18)—No RESUME: Error trapping is handled like a subroutine, with the ON ERROR GOTO instruction sending execution to a specified line number. When the routine is finished, it must return operation to a given point; the RESUME statement furnishes this instruction. The NR message is returned if it is missing.

RW (error code 19)—RESUME without ERROR: RESUME is similar to the RETURN instruction after a GOSUB. If no ON ERROR GOTO instruction has been read by the computer, there is no location stored for a return point.

UE (error code 20)—Unprintable Error: The ERROR instruction, used in conjunction with the error codes, allows us to simulate error conditions for constructing and testing error trapping routines. This code indicates that you are using an illegal number code and the instruction cannot be executed. See page B/1 of the manual for error code numbers.

MO (error code 21)—Missing Operand: To perform a computation we must have some data and be told what to do with it. The symbols which tell us what to do (+, −, * and /) are called operators. The items of data on which these signs operate are called the operands. If this message is returned, you have specified some operation but have not furnished all the data needed to complete the operation.

Top Quality Tape

FD (error code 22)—Bad File Data: This indicates you have received some bad data from an external source, usually the notorious cassette player. This can be especially aggravating because Level II doesn't allow verification of data dumps as it does with program data using the CLOAD? instruction. When large amounts of data are involved it pays to use top quality tape and to make multiple dumps.

L3 (error code 23)—Disk BASIC Error: This error means that an attempt has been made to use a Disk BASIC term without having Disk BASIC in your machine.

Two other messages are sometimes returned which indicate that something is amiss. They are:

Extra Ignored: This is used in conjunction with the INPUT instruction and indicates you have entered too much data. Should you specify loading locations A and B with the INPUT instruction 10 INPUT A,B, and then enter 1,2,3, the computer has no instructions as to what to do with the third item. It loads 1 into A, 2 into B and ignores the data which follows.

REDO: Level I allowed us to enter an expression, such as 12*3 or a variable name, in response to the INPUT prompt. This is not acceptable in Level II. Evaluate all of the needed expressions first and then enter them as simplified values. If it is necessary to get the contents of location X into location Y, do not use the INPUT route, assign the values instead as in 50 Y = X.■

Next month: Error trapping.

✓See List of Advertisers on page 306

All right beginners, let's assemble.

An Idiot's Guide To Assembly Language—Part 2

Robert C. Montgomery
67 Turtle Back Road West
New Canaan, CT 06840

Part I of this article explained the use of Radio Shack's Editor/Assembler program and explored some fundamentals of assembly language programming. The primary purpose of assembly language programming is speed of execution. We promised to draw a football field on the TRS-80's screen so fast that the user would be unable to see it happen.

As a step along the way, we wrote, debugged and executed an exciting program which displays the letter X at screen position 15600, in almost as efficient a manner as could be done by using PRINT @ 240, X in BASIC. The pre-game ceremonies are now complete. Let's get the playing field ready.

In order to print X, we used the following assembly language program:

```
00100           ORG  7F63H      ;STORE AT 32
611=7F63H
00110           LD   A,58H      ;"X"=88=58H
00120           LD   (3CF0H),A  ;SCREEN LOCA
TION 15600=3CF0H
00130 LOOP      JP   LOOP       ;ENDLESS LOOP
00140           END
```

We're going to substitute some TRS-80 graphics characters for X, and put them in other screen locations. Otherwise, we have all the assembly language statements needed to put a gridiron on the field.

We want a double-width sideline, extending from screen position 15744 (PRINT @ 384) through 61 tab positions to screen position 15804. We could write:

```
LD          A,83H        ;CHR$(131)
LD          (3D80H),A    ;SCREEN POSITION 15
744=3D80H
LD          (3D81H),A    ;SCREEN POSITION 15
745=3D81H
LD          (3D82H),A
                         . . . and so on.
```

We would be awfully tired of entering code by the time we finished. A loop is better. In BASIC, we would write:

```
100 FOR I=384 TO 444
110 PRINT @I, CHR$(131);
120 NEXT I
130 PRINT
```

No FOR. . .NEXT routine exists in assembler. Accordingly, we must set up a loop counter. Use register B for that purpose, load it with the value 61, decrement it by one each time the loop is executed, and exit the loop when it is down to zero. Could we have started at zero and exited at 61? Sure; but as you'll see, assembly language contains a convenient test for zero, but none for 61.

We need to increase the screen position by one each time the loop is executed. Load the two byte HL register pair initially with 15744, and it will be up to 15804 when we exit the loop.

We need three new statements:

- DEC B, which means: Decrement the value in the B register by one.
- INC HL, which means: Increment the value in the HL register pair by one.
- JP NZ, (address), which means: Jump to the stated address if the result of the last operation was not zero.

If the last operation was DEC B, the value in the B register would be zero after 61 loops. Thus, if "address" is the place in memory at which the first statement of the loop is stored, the program would execute DEC B, test for zero, and go back to "address" 61 times, before falling through the loop on the 62nd attempt.

One way to figure out what address to use would be to count the machine language codes used prior to the address and set the next one for the first statement in the loop. But, we used a label in our PRINTX program to permit EDTASM to compute this address for us, and we'll do the same here. A label called LOOP1 will show where the 61-iteration loop begins, and LOOP2 will be used to set up an endless loop while we look at the result.

Reload EDTASM, and enter the text appearing in Listing 1. I have changed the ORG address; we're going to need more room.

Assemble the program, record it on cassette, go back to BASIC, set memory size to 32570, load the machine code as NONAME using the System command, and enter / 32571 following the prompt. It works! Or does it? That doesn't look like the right place on the screen. Go back and check the text. Have you found the error? 15744 decimal is 3D80H, not 3C80H.

This is one of the most common errors made in assembly language, other than typos. Check the decimal to hex conversions, and then check them back, hex to decimal. Fortunately, it's easy to fix, using EDTASM's edit routine.

Reload EDTASM and the text (using L). Now let me warn you of another EDTASM idiosyncrasy—tapes don't always load. You set EDTASM up to read the text back in, push all the right buttons (and maybe some wrong ones), and the next thing you know, there are unblinking asterisks and the other signs of a bad load. You push the reset button, right? Wrong! If you do, you'll lose EDTASM, and have to read it back in again. Somehow you have to get EDTASM to cough up an error message. Rewind the tape, and continue the read. Wait until the bad parameters message appears (who thought that one up?), and *then* start your load again.

"EDTASM will also renumber the lines automatically, starting wherever you wish . . ."

Enter E130, space over to the offending C, press C (for change), and put in a D. We can now go on to the other sideline.

Enter D180 and D190. Line 180 was a temporary endless loop for demonstration purposes, and line 190 is no longer the end. The lower sideline is almost a carbon copy of the upper, as shown in Listing 2.

Did you catch *that* error? We forgot to reset the loop counter to 61. Press Break, enter I185, and add line 185 to read:

```
00185     LD     B,3DH     ;61 LOOPS
```

This is like adding a new line to a BASIC program; though it was entered last, it will execute in the proper order. EDTASM will also renumber the lines automatically, starting wherever you wish and using whatever increment you prefer. To do that, enter N100, if you want to start at 100 and increment by the default value of 10. Or, enter N500,100, if you want to count by 100's from line 500.

Cleaning Up The Text

We have a section of code which will draw the upper sideline, and another section which will draw the lower sideline. Let's put descriptive labels on them, and separate them by a few spaces.

Enter I90,1 to insert lines at line 90 with an increment of one. Then enter lines 90 through 92:

```
00090 ;
00091 ;          UPPER SIDELINE
00092 ;
```

Do the same thing before the code for the lower sideline, using new lines for this purpose. Then use N100,10 to renumber everything. After you hit A, the text ought to look like Listing 3.

End Zones

Has it occurred to you that we could have combined Listing I and 2 into a single loop of 61 iterations? Let's try to draw the double-width lines in the end zones in a single routine. One line should start at screen position 15744 (PRINT @ 384), and extend downward for nine screen lines. We want a similar line five tab positions to the right, another 51 tab positions to the right of the second line, and then still another five tab positions to the right of the third.

We can't efficiently INC HL five times, much less 51 times; we need to add to the screen position contained in HL:

```
LD     DE,5H
ADD    HL,DE, which means "Load the value 5 into register pair DE. Then add the contents of DE to whatever is already in register pair HL"
```

We could later add 51 to HL in the same way. Five and 51 are one byte numbers. They can't be stored in register A, and then be written ADD HL,A because if one is a register pair, the other must be also. Could we have written ADD HL,5H, intending to add the value directly to HL? No. Why? Who knows?

Listing 4 demonstrates what we can write. Start with 15744 in HL, add 5, add 51,

```
ØØ1ØØ          ORG     7F3BH          ;STORE AT 32571

ØØ11Ø          LD      A,83H          ;CHR$(131)
ØØ12Ø          LD      B,3DH          ;61 LOOPS
ØØ13Ø          LD      HL,3C8ØH       ;SCREEN POSITION
15744=3C8ØH
ØØ14Ø LOOP1    LD      (HL),A         ;LOAD 131 INTO SCREEN
ØØ15Ø          INC     HL             ;SCREEN POSITION + 1
ØØ16Ø          DEC     B              ;COUNTER - 1
ØØ17Ø          JP      NZ, LOOP1      ;BACK TO 'LOOP1' IF NOT
ZERO
ØØ18Ø LOOP2    JP      LOOP2          ;ENDLESS LOOP
ØØ19Ø          END
```

Listing 1. Top Sideline

```
ØØ18Ø          LD      A,ØBØH         ;CHR$(176)
ØØ19Ø          LD      HL,3F8ØH       ;SCREEN POSITION 16256
ØØ2ØØ LOOP3    LD      (HL),A         ;LOAD 176 INTO SCREEN
ØØ21Ø          INC     HL             ;SCREEN POSITION + 1
ØØ22Ø          DEC     B              ;COUNTER - 1
00230          JP      NZ,LOOP3       ;BACK TO 'LOOP3' IF NOT ZERO
```

Listing 2. Bottom Sideline

✓See List of Advertisers on page 306

"Our amateurish, inefficient program put the football field on the screen so fast you couldn't time it."

add 5, and then add 3. Start another iteration of the loop without changing the contents of HL: 5+51+5+3=64. By adding the final three, we adjusted HL to the value of the screen position which starts the next screen line.

The yard lines on the left side of the field are drawn by the repetitive use of CHR$(170). On the right side, they use CHR$(149). The 50-yard line is double-width: That will be automatic if we use CHR$(170) at tab position 30 and CHR$(149) at tab postion 31. At the top and bottom the yard lines will write over the sidelines. We will therefore need L shaped characters at the point of intersection.

In Listing 4 there was a lot of repetition; ADD HL,DE and LD (HL),A had to be written three times. In BASIC, we would have used a subroutine, and we can do the same thing in assembler:

CALL 2135H, which means: Jump to the subroutine at memory location 2135H, do whatever those instructions say, and return when instructed.

```
ØØ1ØØ ;
ØØ11Ø ;                         UPPER SIDELINE
ØØ12Ø ;
ØØ13Ø            ORG            7F3BH              ;STORE AT 32571
ØØ14Ø            LD             A,83H              ;CHR$(131)
ØØ15Ø            LD             B,3DH              ;61 LOOPS
ØØ16Ø            LD             HL,3D8ØH           ;SCREEN POSITION 15744=3D8
ØH
ØØ17Ø LOOP1      LD             (HL),A             ;LOAD 131 INTO SCREEN
ØØ18Ø            INC            HL                 ;SCREEN POSITION + 1
ØØ19Ø            DEC            B                  ;COUNTER - 1
ØØ2ØØ            JP             NZ,LOOP1           ;BACK TO 'LOOP1' IF NOT ZE
RO
ØØ21Ø ;
ØØ22Ø ;                         LOWER SIDELINE
ØØ23Ø ;
ØØ24Ø            LD             A,ØBØH             ;CHR$(176)
ØØ25Ø            LD             B,3DH              ;61 LOOPS
ØØ26Ø            LD             HL,3F8ØH           ;SCREEN POSITION 16256=3F8
ØH
ØØ27Ø LOOP3      LD             (HL),A             ;LOAD 176 into SCREEN
ØØ28Ø            INC            HL                 ;SCREEN POSITION + 1
ØØ29Ø            DEC            B                  ;COUNTER - 1
ØØ3ØØ            JP             NZ,LOOP3           ;BACK TO 'LOOP3' IF NOT ZE
RO
ØØ31Ø            END

Notes: Statement ØØ31Ø is temporary.
       Add an endless loop at statement ØØ3Ø5 if you want to test.
```

Listing 3. The Complete Sidelines Program

```
ØØ31Ø ;
ØØ32Ø ;                         END ZONES
ØØ33Ø ;
ØØ34Ø            LD             A,ØBFH             ;CHR$(191)
ØØ35Ø            LD             B,9H               ;9 LOOPS
ØØ36Ø            LD             HL,3D8ØH           ;SCREEN POSITION 15744=3D8
ØH
ØØ37Ø LOOP4      LD             (HL),A             ;LOAD 191 INTO SCREEN
ØØ38Ø            LD             DE,5H              ;INCREMENT OF 5
ØØ39Ø            ADD            HL,DE              ;SCREEN POSITION + 5
ØØ4ØØ            LD             (HL),A             ;LOAD 191 INTO SCREEN
ØØ41Ø            LD             DE,33H             ;INCREMENT OF 51
ØØ42Ø            ADD            HL,DE              ;SCREEN POSITION + 51
ØØ43Ø            LD             (HL),A             ;LOAD 191 INTO SCREEN
ØØ44Ø            LD             DE,5H              ;INCREMENT OF 5
ØØ45Ø            ADD            HL,DE              ;SCREEN POSITION + 5
ØØ46Ø            LD             (HL),A             ;LOAD 191 INTO SCREEN
ØØ47Ø            LD             DE,3H              ;INCREMENT OF 3
ØØ48Ø            ADD            HL,DE              ;SCREEN POSITION + 3 (NEW
LINE)
ØØ49Ø            DEC            B                  ;COUNTER - 1
ØØ5ØØ            JP             NZ,LOOP4           ;BACK TO 'LOOP4' IF NOT ZE
RO
```

Listing 4. End Zones

Then write the subroutine instructions, and return by writing:

RET , which means the same as RETURN in BASIC.

Better yet, we can use a label at the point the subroutine begins. Just as in looping, EDTASM will determine its address for us.

We want to use a double loop. The outer loop should provide the seven iterations necessary to draw the vertical yard lines and the inner loop should provide the five iterations needed to draw five lines. Use register C as an additional loop counter. Listing 5 contains the text.

The yard lines on the right half of the screen are a mirror image of those on the left; the only other task is to join the yard lines to the sidelines. We need L shaped characters at the intersection of the yard lines with the sidelines: CHR$(171) in the upper left, CHR$(186) in the upper right, CHR$(151) in the lower left, and CHR$(181) in the lower right.

Finally, place an endless loop at the end of the program so we can see the results on the screen. Listing 6 shows the complete program after assembly.

We have noted a few programming techniques which might have been done in easier ways, by combining loops, and through use of subroutines. There are *many* other improvements we might have made to save operations and key strokes. Show this program to an accomplished assembly language programmer: he'll tell you how bad it really is.

Let's run it, bad though it may be. Record the machine code (after using A) and then use W to record the text. Exit EDTASM, using B. Enter 32570 as the memory size, and load NONAME, using the system command. When the prompt appears, get your stopwatch ready to time it (BASIC took 3.4 seconds, you'll recall). Now enter /32571 to run the assembly language version.

That stopwatch wasn't much help, was it? Our amateurish, inefficient program put the football field on the screen so fast you couldn't time it. If you want speed, this is the way to get it!

Memory Requirements

Go back to Listing 6. Note the first column of numbers; these are the hex representations of the places in memory in which this program is stored. The first is 7F3BH, or 32571 decimal. The program starts there because we told it to, using ORG. Look at the second column. These are the hex representations of the data stored at these memory locations. Some are instructions; some, especially those requiring two byte numbers, require three codes to store. Some powerful instructions require only one byte. The memory locations in the first

"How do we know what numbers to POKE and where to POKE them?"

```
00510 ;
00520 ;
00530 ;
00540           LD      A,0AAH          ;CHR$(170)
00550           LD      B,7H            ;7 OUTER LOOPS (EXCLUDING
TOP AND BOTTOM SIDELINES)
00560           LD      HL,3DCAH        ;SCREEN POSITION 15818=eDC
AH
00570 ;
00580 ;                 OUTER LOOP
00590 ;
00600 LOOP6     LD      DE,5H           ;INCREMENT OF 5
00610           LD      C,5H            ;5 INNER LOOPS
00620 ;
00630 ;                 INNER LOOP
00640 ;
00650 LOOP7     CALL    SUBRT           ;SUBROUTINE
00660           DEC     C               ;INNER COUNTER - 1
00670           JP      NZ,LOOP7        ;BACK TO 'LOOP7' IF NOT ZE
RO
00680           LD      DE,27H          ;INCREMENT OF 39
00690           ADD     HL,DE           ;SCREEN POSITION + 39 (NEW
LINE)
00700           DEC     B               ;OUTER COUNTER - 1
00710           JP      NZ,LOOP6        ;BACK TO 'LOOP6' IF NOT ZE
RO

        .... and, later:

01360 ;
01370 ;                 SUBROUTINE
01380 ;
01390 SUBRT     LD      (HL),A          ;LOAD CHARACTER INTO SCREE
N
01400           ADD     HL,DE           ;SCREEN POSITION + INCREME
NT
01410           RET                     ;BACK TO INSTRUCTION FOLLO
WING CALL
```

Listing 5. Left Yardlines

column increase by one, two or three, depending on the number of bytes required by the previous instruction.

Some instructions require no code (ORG and END are examples). The labels used are listed at the end of the assembly, each with its computed location in memory. These are not translated into code: all are used in assembly, but neither they, nor the line numbers, nor explanatory comments, actually require memory space after assembly. The last memory location is 7FEAH, or 32746 decimal, which means the entire inefficient program will require 176 bytes of machine code. In contrast, the BASIC program shown in Part I required 374 bytes.

Using the Program as a BASIC Subroutine

Now we have a machine language program; how do we hook it up to a BASIC program?

Reload the text using EDTASM, and replace lines 1330 and 1350 as follows:

```
01330 ;         RETURN TO BASIC
01350   RET                     ;RETURN TO BASIC
```

CALL will be part of the BASIC program.

Record the revised program after reassembly, exit EDTASM, and load NONAME as before, being careful to set memory size to 32570. Enter the BASIC program shown in Listing 7.

Run it—it works. Those two POKEs put the starting address of the machine language subroutine into the BASIC monitor (as usual, it's done the hard way). The starting address was 32571, or 7F3BH, a two byte number in hex. The most significant byte—the one that really counts in evaluating the size of the number—is 7F. The least significant byte is 3B. Now reverse them (don't ask me why) and you get 3B, 7F. Translate them into decimal and you get 59, 127, which is what you POKEd into 16526 and 16527. That's all you do to set up the starting address.

To call the subroutine, write X = USR(0) and it's done. Note also that we finally cleared the screen in line 100, and that line 170 prevents the screen from scrolling when the end of the program is encountered.

Using Data Statements Instead of System

Loading machine code through the System command may be confusing to users of your football game. To make it easier for them, there's another way of mak-

"Programs expand to fill the memory available for their storage."

ing things harder for you. You can POKE all that code into the required places in memory, using the BASIC statements shown in Listing 8.

How do we know what numbers to POKE and where to POKE them? The machine code starts at memory location 32571. In the original version, which contained an endless loop, it ended at 32747. After modification, it ends at 32744. While the machine code is still in memory (after a system load), write the following:

```
FOR I = 32571 TO 32744 : PRINT I; PEEK(I) : NEXT
```

You'll see 174 numbers flash by, in 174 locations. The numbers are the decimal representations of the machine code, and they are the same numbers, in the same order, as those which appear in the data statements of Listing 8.

There is an easy way to check your work. Before you run the program in Listings 7 and 8, change statement 30 temporarily to read:

```
30 READ X : PRINT I, X, PEEK(I) : NEXT
```

When you run this modified program, you'll see three columns of numbers flash by:

- The memory location (I).
- The value read from the data statement (X).
- The machine code entered earlier through the system command (PEEK (I)).

If the value from the data statement doesn't equal the machine code value, you've obviously got a problem in the memory location. Don't be fooled by the fact that the program still operates correctly; it's still using the machine code entered earlier. Check each of the 174 codes. If they're correct, change line 30 back to its original form, and run. The codes are now coming from the data statements.

The memory size must be set at 32570 when the TRS-80 is powered up. After that, CLOAD proceeds in the usual manner.

Saving Memory

We're using a lot of memory—more, even, than the original BASIC program we used to draw the same graphics. We must protect a minimum of 174 bytes of high memory to store and operate the subroutine. Lines 10 through 70 in Listing 8 require an additional 628 bytes. This may be unimportant, but more likely you're subject to Parkinson's Law of Programming[1], and you need every byte you can find.

By adding a line to the code in Listing 8, we can save 628 bytes. Try this:

```
99 DELETE 10-99
```

Run it: All you get is a READY and a prompt. Run again—there's our field. Now LIST—lines 100 through 170 are all that remain (the machine code must still be in 32571 through 32744, or there would be no field).

What happened? Lines 10 through 70 were executed in the usual way, with 174 bytes POKEd into high memory. When the computer encountered DELETE 10–99, it did as it was told, and lines 10 through 99 self-destructed. The program then halted, awaiting instructions from the keyboard. You told it to run, and it did—beginning at

Listing 6. Complete Program

```
*A

                ØØ1ØØ ;
                ØØ11Ø ;                 UPPER SIDELINE
                ØØ12Ø ;
7F3B            ØØ13Ø        ORG        7F3BH              ;STORE AT 32
71=7F3BH
7F3B 3E83       ØØ14Ø        LD         A,83H              ;CHR$(131)
7F3D Ø63D       ØØ15Ø        LD         B,3DH              ;61 LOOPS
7F3F 218Ø3D     ØØ16Ø        LD         HL,3D8ØH           ;SCREEN POSI
TION 15744=3D8ØH
7F42 77         ØØ17Ø LOOP1  LD         (HL),A             ;LOAD 131 IN
TO SCREEN
7F43 23         ØØ18Ø        INC        HL                 ;SCRFEN POSI
TION + 1
7F44 Ø5         ØØ19Ø        DEC        B                  ;COUNTER - 1

7F45 C2427F     ØØ2ØØ        JP         NZ,LOOP1           ;BACK TO 'LO
OP1' IF NOT ZERO
                ØØ21Ø ;
                ØØ22Ø ;                 LOWER SIDELINE
                ØØ23Ø ;
7F48 3EBØ       ØØ24Ø        LD         A,ØBØH             ;CHR$(176)
7F4A Ø63D       ØØ25Ø        LD         B,3DH              ;61 LOOPS
7F4C 218Ø3F     ØØ26Ø        LD         HL,3F8ØH           ;SCREEN POSI
```

Program continues

✓See List of Advertisers on page 306

```
TION + 1
7F4F 77         ØØ27Ø LOOP3   LD      (HL),A              ;LOAD 176 IN
TO SCREEN
7F5Ø 23         ØØ28Ø         INC     HL                  ;SCREEN POSI
TION + 1
7F51 Ø5         ØØ29Ø         DEC     B                   ;COUNTER - 1
7F52 C24F7F     ØØ3ØØ         JP      NZ,LOOP3            ;BACK TO 'LO
OP3' IF NOT ZERO
                ØØ31Ø ;
                ØØ32Ø ;               END ZONES
                ØØ33Ø ;
7F55 3EBF       ØØ34Ø         LD      A,ØBFH              ;CHR$(191)
7F57 Ø6Ø9       ØØ35Ø         LD      B, 9H               ;9 LOOPS
7F59 218Ø3D     ØØ36Ø         LD      HL,3D8ØH            ;SCREEN POSI
TION 15744=3D8ØH
7F5C 77         ØØ37Ø LOOP4   LD      (HL),A              ;LOAD 191 IN
TO SCREEN
7F5D 11Ø5ØØ     ØØ38Ø         LD      DE,5H               ;INCREMENT O
F 5
7F6Ø 19         ØØ39Ø         ADD     HL,DE               ;SCREEN POSI
TION + 5
7F61 77         ØØ4ØØ         LD      (HL),A              ;LOAD 191 IN
TO SCREEN
7F62 1133ØØ     ØØ41Ø         LD      DE,33H              ;INCREMENT O
F 51
7F65 19         ØØ42Ø         ADD     HL,DE               ;SCREEN POSI
TION + 51
7F66 77         ØØ43Ø         LD      (HL),A              ;LOAD 191 IN
TO SCREEN
7F67 11Ø5ØØ     ØØ44Ø         LD      DE,5H               ;INCREMENT O
F 5
7F6A 19         ØØ45Ø         ADD     HL,DE               ;SCREEN POSI
TION + 5
7F6B 77         ØØ46Ø         LD      (HL),A              ;LOAD 191 IN
TO SCREEN
7F6C 11Ø3ØØ     ØØ47Ø         LD      DE,3H               ;INCREMENT O
F 3
7F6F 19         ØØ48Ø         ADD     HL,DE               ;SCREEN POSI
TION PLUS 3 (NEW LINE)
7F7Ø Ø5         ØØ49Ø         DEC     B                   ;COUNTER - 1
7F71 C25C7F     ØØ5ØØ         JP      NZ,LOOP4            ;BACK TO 'LO
OP4' IF NOT ZERO
                ØØ51Ø ;
                ØØ52Ø ;               YARDLINES (LEFT HALF)
                ØØ53Ø ;
7F74 3EAA       ØØ54Ø         LD      A,ØAAH              ;CHR$(17Ø)
7F76 Ø6Ø7       ØØ55Ø         LD      B,7H                ;7 OUTER LOO
PS (EXCLUDING TOP AND BOTTOM SIDELINES)
7F68 21CA3D     ØØ56Ø         LD      HL,3DCAH            ;SCREEN POSI
TION 15818=3DCAH
                ØØ57Ø ;
                ØØ58Ø ;               OUTER LOOP
                ØØ59Ø ;
7F7B 11Ø5ØØ     ØØ6ØØ LOOP6   LD      DE,5H               ;INCREMENT O
F 5
7F7E ØEØ5       ØØ61Ø         LD      C,5H                ;5 INNER LOO
PS
                ØØ62Ø ;
                ØØ63Ø ;               INNER LOOP
                ØØ64Ø ;
7F8Ø CDE87F     ØØ65Ø ; LOOP7 CALL    SUBRT               ;SUBROUTINE
7F83 ØD         ØØ66Ø         DEC     C                   ;INNER COUNT
ER - 1
7F84 C28Ø7F     ØØ67Ø         JP      NZ,LOOP7            ;BACK TO 'LO
OP7' IF NOT ZERO
7F87 1127ØØ     ØØ68Ø         LD      DE,27H              ;INCREMENT O
F 39
7F8A 19         ØØ69Ø         ADD     HL,DE               ;SCREEN POSI
TION + 39 (NEW LINE)
7F8B Ø5         ØØ7ØØ         DEC     B                   ;OUTER COUNT
ER - 1
7F8C C27B7F     ØØ71Ø         JP      NZ,LOOP6            ;BACK TO 'LO
OP6' IF NOT ZERO
                ØØ72Ø
                ØØ73Ø ;               YARDLINES (RIGHT HALF)
                ØØ74Ø ;
7F8F 3E95       ØØ75Ø         LD      A,95H               ;CHR$(149)
7F91 Ø6Ø7       ØØ76Ø         LD      B,7H                ;7 OUTER LOO
PS (EXCLUDING TOP AND BOTTOM SIDELINES)
7F93 21DF3D     ØØ77Ø         LD      HL,3DDFH            ;SCREEN POSI
TION 15839=3DDFH
                ØØ78Ø ;
```

Program continues

✓See List of Advertisers on page 306

"Don't expect to read the EDTASM manual as you would a detective novel."

the first remaining line.

You can use a controlled crash like this to be doubly sure that the user remembered to set the memory size. Add the following lines to the program:

```
80 CLS : PRINT "IF 'MEMORY SIZE?' WAS ANSWERED
   '32570',": PRINT "TYPE 'RUN' AND PRESS 'ENTER'"
90 PRINT "OTHERWISE, START AGAIN"
```

This, of course, will also self-destruct, and the total memory requirement for the graphics routine is now 174 bytes, plus the controls in lines 140 through 160.

If you want to go on in machine language, you're now ready to read and understand any of the several excellent texts available at your local computer store. Don't expect to read the EDTASM manual as you would a detective novel; it wasn't really intended as an instruction text in the first place.

I think you'll *want* to go further, if for no other reason than to save yourself trouble by using more powerful instructions. You can use assembly language without knowing very much about it. And the way to learn a *lot* is to begin by learning a *little.*■

¹Programs expand to fill the memory available for their storage.


```
                  ØØ79Ø ;                      OUTER LOOP
                  ØØ8ØØ ;
7F96 11Ø5ØØ       ØØ81Ø LOOP8     LD           DE,5H               ;INCREMENT O
F 5
7F99 ØEØ5         ØØ82Ø           LD           C,5H                ;5 INNER LOO
PS
                  ØØ83Ø ;
                  ØØ84Ø ;                      INNER LOOP
                  ØØ85Ø ;
7F9B CDE87F       ØØ86Ø LOOP9     CALL         SUBRT               ;SUBROUTINE
7F9E ØD           ØØ87Ø           DEC          C                   ;INNER COUNT
ER - 1
                  ØØ88Ø
7F9F C29B7F       ØØ89Ø           JP           NZ,LOOP9            ;BACK TO 'LO
OP9' IF NOT ZERO
7FA2 1127ØØ       ØØ9ØØ           LD           DE,27H              ;INCREMENT O
F 39
7FA5 19           ØØ91Ø           ADD          HL,DE               ;SCREEN POSI
TION + 39 (NEW LINE)
7FA6 Ø5           ØØ92Ø           DEC          B                   ;OUTER COUNT
ER - 1
7FA7 C2967F       ØØ93Ø           JP           NZ,LOOP8            ;BACK TO 'LO
OP8' IF NOT ZERO
                  ØØ94Ø ;
                  ØØ95Ø ;                      L-SHAPES
                  ØØ96Ø ;                      UPPER LEFT
                  ØØ97Ø ;
7FAA 3EAB         ØØ98Ø           LD           A,ØABH              ;CHR$(171)
7FAC Ø6Ø5         ØØ99Ø           LD           B,5H                ;5 LOOPS
7FAE 11Ø5ØØ       Ø1ØØØ           LD           DE,5H               ;INCREMENT O
F 5
7FB1 218A3D       Ø1Ø1Ø           LD           HL,3D8AH            ;SCREEN POSI
TION 15754=3D8AH
7FB4 CDE87F       Ø1Ø2Ø LOOPA     CALL         SUBRT               ;SUBROUTINE
7FB7 Ø5           Ø1Ø3Ø           DEC          B                   ;COUNTER - 1
7FB8 C2B47F       Ø1Ø4Ø           JP           NZ,LOOPA            ;BACK TO 'LO
OPA' IF NOT ZERO
                  Ø1Ø5Ø ;
                  Ø1Ø6Ø ;                      LOWER LEFT
                  Ø1Ø7Ø ;
7FBB 3EBA         Ø1Ø8Ø           LD           A,ØBAH              ;CHR$(186)
7FBD Ø6Ø5         Ø1Ø9Ø           LD           B,5H                ;5 LOOPS
7FBF 218A3F       Ø11ØØ           LD           HL,3F8AH            ;SCREEN POSI
TION 16266=3F8AH
Note:  Did you notice line 880?  EDTASM will keep a line number even if
       no text follows.  It does no harm.
7FC2 CDE87F       Ø111Ø LOOPB     CALL         SUBRT               ;SUBROUTINE
7FC5 Ø5           Ø112Ø           DEC          B                   ;COUNTER - 1
7FC6 C2C27F       Ø113Ø           JP           NZ,LOOPB            ;BACK TO 'LO
OPB' IF NOT ZERO
                  Ø114Ø ;
                  Ø115Ø ;                      UPPER RIGHT
                  Ø116Ø ;
7FC9 3E97         Ø117Ø           LD           A,97H               ;CHR$(151)
7FCB Ø6Ø5         Ø118Ø           LD           B,5H                ;5 LOOPS
7FCD 219F3D       Ø119Ø           LD           HL,3D9FH            ;SCREEN POSI
TION 15775=3D9FH
7FDØ CDE87F       Ø12ØØ LOOPC     CALL         SUBRT               ;SUBROUTINE
7FD3 Ø5           Ø121Ø           DEC          B                   ;COUNTER - 1
7FD4 C2DØ7F       Ø122Ø           JP           NZ,LOOPC            ;BACK TO 'LO
OPC' IF NOT ZERO
                  Ø123Ø ;
                  Ø124Ø ;                      LOWER RIGHT
                  Ø125Ø ;
7FD7 3EB5         Ø126Ø           LD           A,ØB5H              ;CHR$(181)
7FD9 Ø6Ø5         Ø127Ø           LD           B,5H                ;5 LOOPS
7FDB 219F3F       Ø128Ø           LD           HL,3F9FH            ;SCREEN POSI
TION 16287=3F9FH
7FDE CDE87F       Ø129Ø LOOPD     CALL         SUBRT               ;SUBROUTINE
7FE1 Ø5           Ø13ØØ           DEC          B                   ;COUNTER - 1
7FE2 C2DE7F       Ø131Ø           JP           NZ,LOOPD            ;BACK TO 'LO
OPD' IF NOT ZERO
                  Ø132Ø ;
```

Program continues

```
               Ø133Ø ;                         ENDLESS LOOP
               Ø134Ø ;
7FE5 C3E57F    Ø135Ø ; LOOPF    JP             LOOPF              ;ENDLESS LOOP
               Ø136Ø ;
               Ø137Ø ;                         SUBROUTINE
               Ø138Ø ;
7FE8 77        Ø139Ø   SUBRT    LD             (HL),A             ;LOAD CHARAC
TER INTO SCREEN
7FE9 19        Ø14ØØ            ADD            HL,DE              ;SCREEN POSI
TION + INCREMENT
7FEA C9        Ø141Ø            RET                               ;BACK TO INS
TRUCTION FOLLOWING CALL
ØØØØ           Ø142Ø            END
ØØØØØ TOTAL ERRORS

LOOPF    7FE5
LOOPD    7FDE
LOOPC    7FDØ
LOOPB    7FD2
LOOPA    7FB4
LOOP9    7F9B
   ....
   ....
LOOP1    7F42
READY CASSETTE
```

```
1ØØ CLS
11Ø PRINT "HIT ANY KEY TO DRAW THE FIELD"
12Ø A$=INKEY$
13Ø IF A$="" THEN 12Ø
14Ø POKE 16526,59
15Ø POKE 16527,127
16Ø X = USR(Ø)
17Ø GOTO 17Ø
```

Listing 7. BASIC Program

```
1Ø POKE 16553,255 : REM -- CANCELS AUTO-RESTORE IN SOME TRS-8Ø'S
2Ø FOR I=32571 TO 32744
3Ø READ X : POKE I,X : NEXT
5Ø DATA 62,131,6,61,33,128,61,119,35,5,194,66,127,62,176,6,61,33,
128,63,119,35,5,194, 79,127,62,191,6,9,33,128,61,119,17,5,Ø,25,119,
17,51,Ø,25,119,17,5,Ø,25,119,17,3,Ø,25,5,194,92,127,62,17Ø,6,7,33,2Ø2,61,
17,5,Ø,14,5,2Ø5,23Ø,127,13,194,128,127
60 DATA 17,39,Ø,25,5,194,123,127,62,149,6,7,33,223,61,17,5,Ø,14,5,
2Ø5,23Ø,127,13,194,155,127,17,39,Ø,25,5,194,15Ø,127,62,171,6,5,17,5,
Ø,33,128,61,2Ø5,23Ø,127,5,194,18Ø,127,62,186,6,5,33,138,63,2Ø5,23Ø,
127,5,194,194,127,62,151,6,5,33,159,61
7Ø DATA 2Ø5,23Ø,127,5,194,2Ø8,127,62,181,6,5,33,159,63,2Ø5,23Ø,127,
5,194,222,127,2Ø1,119,25,2Ø1
```

Listing 8. Code to POKE Subroutine into Memory

Suburban police departments computerize, then network, in their fight against crime.

Dragnetwork

Alicia Kennedy
6030 N. Sheridan Road
Suite 1906
Chicago, IL 60660

Why a microcomputer network? Perhaps when the chief of a small Illinois police department purchased his TRS-80 Model I, he had not thought of the possibility of a network.

Ironically, about a year later, through personal determination and exploration, he and several neighboring departments found themselves joined together to form what is believed to be one of the few suburban microcomputer networks in the country. Presently the chief is active in educating other department audiences on the how-to's of the microcomputer communications networks.

Originally purchased as the department's "Boy Wonder," the 80 was intended to alleviate administrative procedures. The department later discovered that pertinent files could be maintained to graph and plot crime tendencies in addition to tracking down offenders.

Word of the project has spread to surrounding police departments, who have since become the elements of a microcomputer network, joined by phone modem. Ultimately, about eight Du Page County Illinois police departments hope to join the network.

Though the I.B.M.s and Honeywells may shine in the blinding path of computer technology, the microcomputer could be considered the prodigy of the computer sensation. When the Hanover Department decided to expand their system, they were able to accomodate their own needs while also staying within the department's minimal budget.

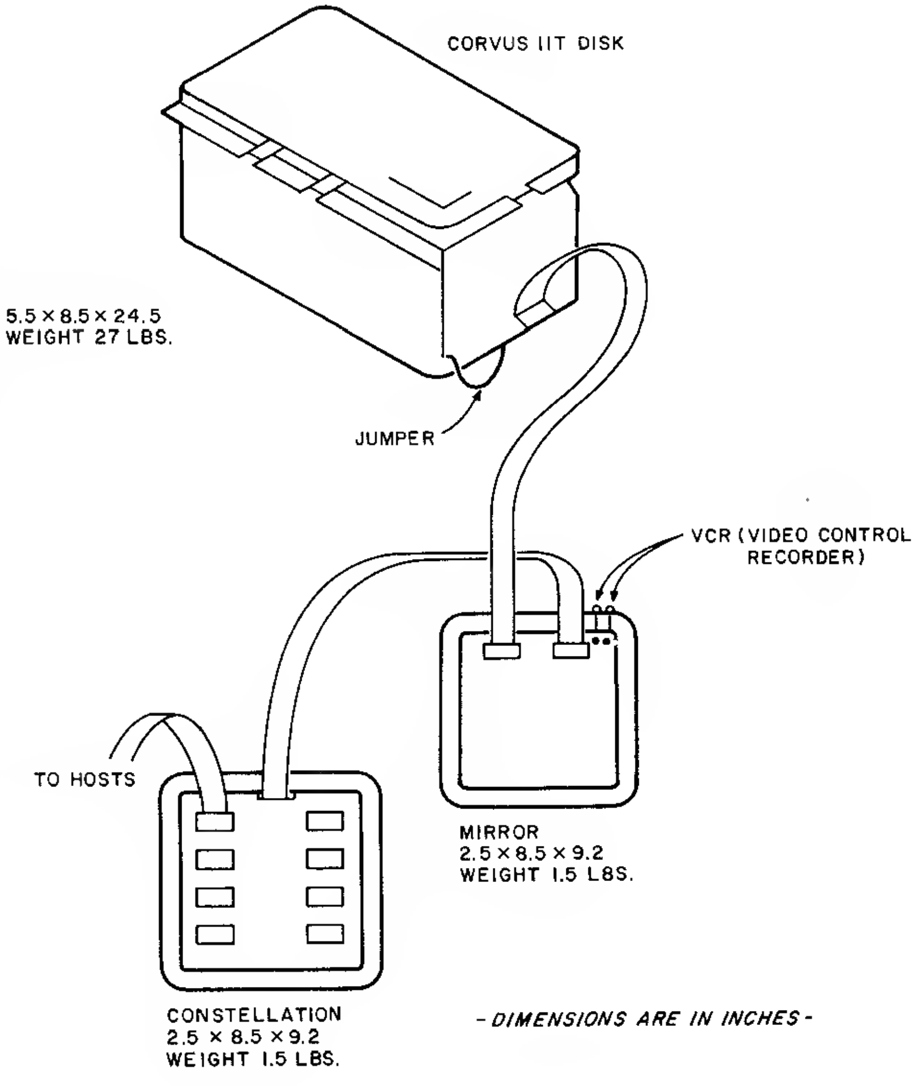


Fig. 1

Components and Cost

This network, constructed by the police department, uses a host multiplexor (a device that can provide multiple I/O to a computer) and a disk drive. Installing the Corvus IIT Disk Drive and Constellations proved to be a wise decision. The multiplexor enables two or more microprocessors to share the same data base simultaneously. The data base, or drive in this case, stores ten million

Fastload

FOR TRS-80* MODEL I USERS ONLY

16 Times Normal Speed

*TRS-80 is a trademark of Tandy Corp.

- High speed load TRS-80* Level II cassettes
- Input 15K byte Level II program in 15 seconds
- Search BASIC or SYSTEM programs by name

Unlike other high speed tape input devices, FASTLOAD uses standard format cassettes. Therefore, there is no need to re-record on other media. At 8000 baud, FASTLOAD is faster than disk for short programs. FASTLOAD reads tapes at the fast-forward speed of the CTR-41 cassette recorder. The recorder can also be used for CSAVE at the normal speed.

FASTLOAD connects to the 40 pin I/O or to the Expansion box. The control program does not use computer memory because it is in a built-in PROM. Other valuable features are keyboard debounce program, automatic key repeat routine and key-beep via cassette speaker. Price is $188.00 for FASTLOAD and $95.00 for the modified CTR-41 recorder.

✓422

Personal Micro Computers Inc.

475 Ellis Street, Mountain View, CA 94043 (415) 968-1604

✓See List of Advertisers on page 306

bytes of information—all of which are accessible within seconds to each and every computer involved (The data rate transfer is a little over 50 kilobytes per second.).

The Constellation's eight ports will allow up to eight computers to feed into the disk drive. However, eight Constellations can be chained to a single Corvus Drive allowing up to 64 computer connections. The drive interfaces to the TRS-80 via a card and an expansion cable.

The Constellation and drive give maximum performance in a minimal space (a little over a cubic foot for both). Chips in the Constellation decipher the various departments and allow each computer to talk to the drive. Information is transferred within a matter of seconds, the Constellation acting as a core to which microprocessors are connected in a "celestial" fashion. Four drives may be connected in any combination of 10 and 20 megabytes.

Enlisting components like the Corvus Drive and Constellation may prove to save thousands. The Drive is equivalent to 94 Radio Shack disk drives, each of which cost $499. Even if you could fit 94 disk drives to the Model I, it would cost approximately $46,000. The 10 megabyte Corvus Drive lists for approximately $5,350. The Constellation, which allows multiple entry on all programs stored on the disk, lists at a price of $750.

Twenty megabyte drives are available for approximately $6,400.

Should the disk become filled, there is another device, the Corvus Mirror, which inexpensively stores 100 Megabytes of information. The Mirror uses a two-hour video cassette tape. Ten million bytes of information can be transferred in less than 15 minutes. Besides its backup capability, the Mirror can be used for retrieving any file or program within minutes, without operator intervention.

It's apparent that the small businessman will no longer have to wonder how to afford microcomputer precision through interfacing. Microcomputer networks will become as common as the calculator. ■

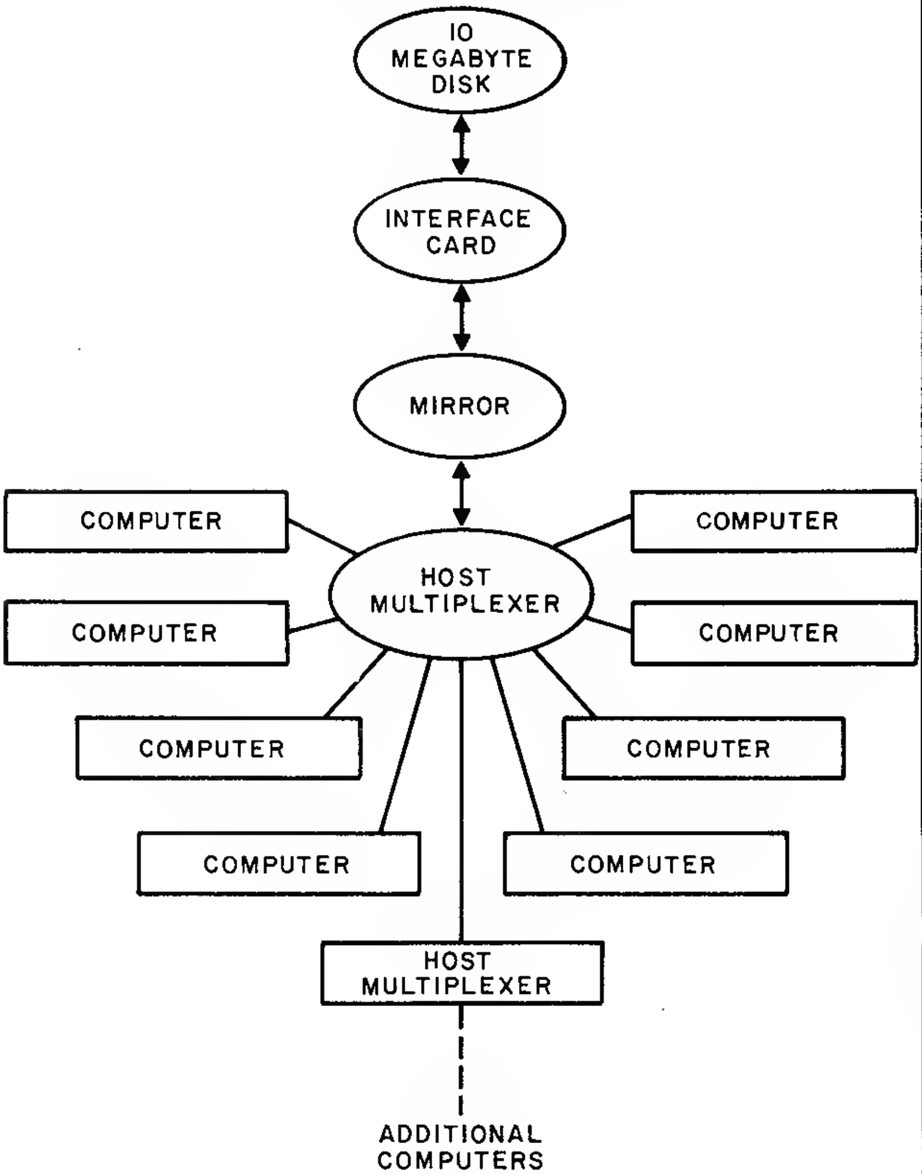
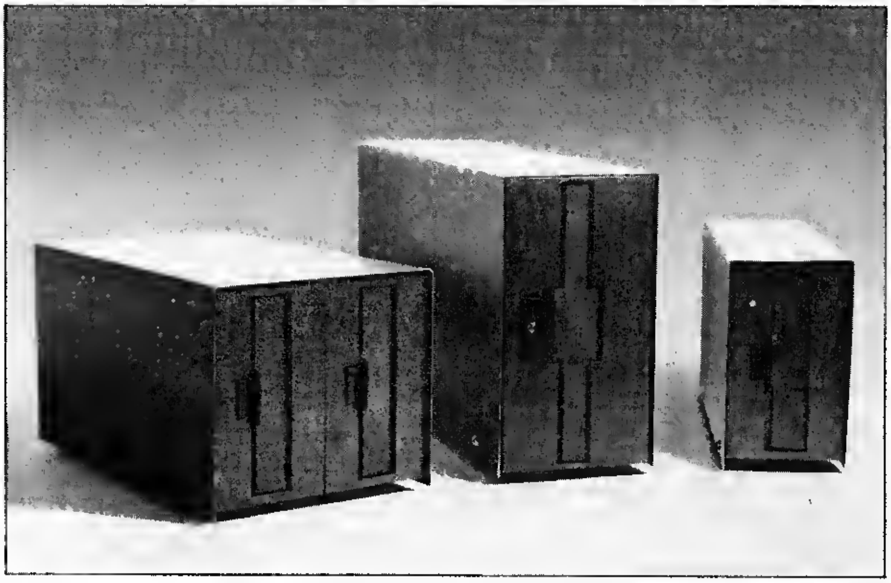


Fig. 2



More power to you.

Disk drives, hardware and software— now more affordable *and* more available!

NOW IN STOCK! MODEL III DISK CONTROLLER BOARDS!

All retrofit packages include circuitry, power supplies, mounting brackets, cables and everything required to turn your Model III into a reliable, disk-based system. 100% compatible with Radio Shack hardware!

AM-1 — Supports 4 single-sided, double-density 5¼-inch disk drives **$445**

AM-2 — Supports 4 double-sided, double-density 5¼-inch disk drives **$545**

AM-3 — Supports 2 internal 5¼-inch single-sided, double-density disk drives and 2 external 8-inch single-sided, double-density disk drives **$645**

TRS-80™ SOFTWARE

MAKE 80©
Converts standard 35 or 40 track diskettes into 80-track-readable diskettes **$14.95**

Model III TRSDOS
40-track Model III TRSDOS & manual **$20**
80-track Model III TRSDOS & manual **$35**

Attention OEM's & Distributors:
Call us for details on our attractive pricing!

New Store Hours:
Monday-Friday 10-6, Sat. 10-5

™TRS-80 is a trademark of Tandy Corp.

SPECIALS!

SAVE ON COMPLETE DISK DRIVE PACKAGES!

5¼-inch drives
40-track MPI 51 with case, power supply and extender cables **$320**
80-track MPI 91 with case, power supply and extender cable **$425**
Dual-headed 80-track drive with case, power supply **$550**

8-inch drives
Single Siemens 8-inch drive with single case & power supply **$695**

TEAC DISK DRIVES— THE BEST WE'VE SEEN TO DATE!

Teac 40-track disk drive
$299 Complete!
Teac 80-track disk drive
$399 Complete!

PRINTER SPECIAL!

Epson MX-80 Printer
$495 Cash Discounted

The power behind the drives©

452

A.M. ELECTRONICS, INC.
3366 Washtenaw Ave.
Ann Arbor, Michigan 48104
(313) 973-2312

NEW TRS-80™ MODEL III! NOW WITH DISK DRIVES!

Model III with dual 80-track disk drives—704K of reliable disk storage on only two drives!
- Features two double-density 80-track drives
- 48K RAM
- Includes A.M. Electronics controller board
- Complete, ready-to-run
  Now only **$2295!**

The Ultimate Small Business Computer: Model III with four dual-headed 80-track disk drives!
- Features four drives
- 48K RAM
- 2.8 Megabytes reliable disk storage
- Includes A.M. Electronics controller board
- Complete, ready-to-run
  Now only **$3,795!**

CASES & POWER SUPPLIES

5½-inch enclosures
Single drive unit case and power supply..................... **$85**
Dual drive unit case and power supply..................... **$120**
(Extender cables $15 each extra)

8-inch enclosures
Single drive unit case and power supply................... **$150**

Send For Your FREE Disk Drive Guide!
Get your concise and fact-filled guide from A.M. Electronics, written in engaging Question/Answer style. Call or write us to order your Guide today!

See List of Advertisers on page 306

An 80 runs the night shift in an astronomical observatory.

Application in Real Time

Russell M. Genet
Fairborn Observatory
1247 Folk Rd.
Fairborn, OH 45324

Some of you probably work in an environment where you monitor a complex, real-time control task that can be more efficiently handled by a computer. While some organizations remain in the dark ages, monitoring such tasks with relays and mechanical timers, many have discovered dedicated microprocessors with their high speed machine language controls stored in ROM.

Still others have discovered microcomputers, such as the TRS-80, that can handle control requirements in BASIC without resorting to machine language at all.

While I'd love to show you how I use an '80 to control my oil refinery in Houston, you'll have to settle for my application at the Fairborn Observatory, where it helps me take photoelectric measurements of the color and brightness of variable stars.

Interface

Since the TRS-80 can't be directly connected to machinery, nor does it provide for remote communication to an operator, some sort of interface is required.

The ideal interface would be low in cost, have just a few chips, be convenient to modify, easy to check for proper operation via built-in indicators, and would come in a nice looking cabinet. Also, it ought to handle the remote communications, have a built-in clock, and relieve the TRS-80 of mundane tasks like counting.

Fortunately, the state of the art in programmable peripheral LSI interface chips is such that these requirements are easily met with an interface of just a half dozen chips. The port structure of the TRS-80 makes this job straightforward, and the peripheral chips can be programmed and controlled in BASIC. The data lines on all the good LSI peripheral chips are tri-stated, so that they can be connected to the TRS-80's bus without buffers. In the smaller control interfaces, address decoding is not required, as address lines A4–A7 (after inverting) can be used directly as the $\overline{CS}$ signals for the LSI chips.

To interconnect the chips within the interface, one could, of course, develop a PC board, run off a thousand, keep one and toss 999. But, for this project there is really nothing wrong with sticking some breadboard sockets together or wirewrapping on a perf board, as long as one doesn't forget to put it all in a nice looking cabinet!

A 40 pin DIP jumper to the breadboard interface on one side, and an edge card connector to the computer on the other joins the two; another DIP jumper and long cable connect the interface to the operators and machines.

Photos by Dan Hollingsworth

Photo 1. The remote electronics, photometer with telescope, at night, at the Fairborn Observatory. The author especially likes to observe eclipsing binary stars. The photometer head includes wide angle eyepiece at top, microscope eyepiece in the middle and photo multiplier enclosure at bottom.

In the Observatory

The initial system at the observatory was entirely manual. As observations were made, starlight output was recorded on a strip chart recorder. Annotations were written by hand to identify the star, time, and filter color. Not only did this use up clear weather observing time (a very scarce resource in Ohio), but, the next morning, it actually took longer to read the strip chart and make the required data reductions than it did to make the observations the previous night.

When the backlog of unreduced strip charts piled up in a run of good weather, I used to hope for bad weather just so I'd have a chance to catch up.

After some thought, I decided that here was a situation for a real-time computer.

I rated all the available computers against pre-established criteria. I was almost disappointed when the TRS-80 was the clear winner. Who would believe that something you could buy at the corner store was the best choice for this exotic, astronomical, control task venture? Except for its inability to stand out from the computer crowd, I never regretted the choice and enjoyed a surprise dividend in terms of local experts and readily available equipment.

In the interface I designed, an Intel 8255 Programmable Peripheral Interface (PPI)

LSI chip controls input from the keypad encoder. It also issues and receives timing and other commands using the bit set/reset features of Port C. Two Intel 8253 Programmable Interval Timers (PIT) LSI chips provide the precision timing signals, control counters and timers.

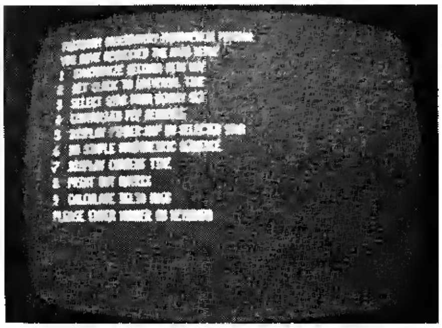


Photo 2. The main menu shows the functions the observer can select. Once a function is completed, control automatically goes back to the main menu so another function can be selected.

Each PIT chip contains three, independently controlled, computer set, 16 bit counters. Each counter can be operated in one of five different modes.

The system is relatively ediot-proof. As darkness falls, the operating program is loaded into the computer, and the month and Julian date are also entered. The video monitor in the computer room is turned off, along with the lights, and the door is locked for the night. (The first night, I sneaked a peek to see if the printer was really working.)

Meanwhile, out in the observatory, once the remote video monitor warms up, the main menu is displayed. As the seconds tick in the clock, the interface is set against the tick from the National Bureau of Standards radio station WWV. This is exact to within a thousandth of a second. Once the seconds are synchronized with WWV, the program returns to the main menu and the clock is then set "at the tone."

The first star pair is selected from a displayed list and a controlled sequence

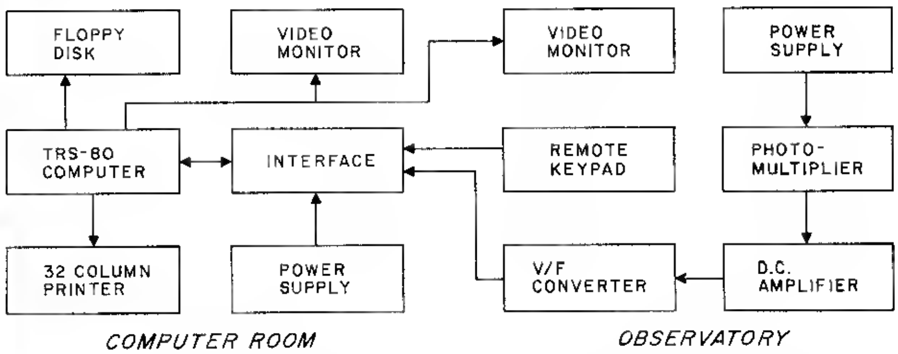


Fig. 1. The small interface ties the TRS-80 to the observatory some 60 feet away. A hand held hexadecimal keypad and remote video monitor provide observer/computer communications. Starlight falling on the photomultiplier is amplified and converted to digital pulses in the voltage-to-frequency (V/F) converter.

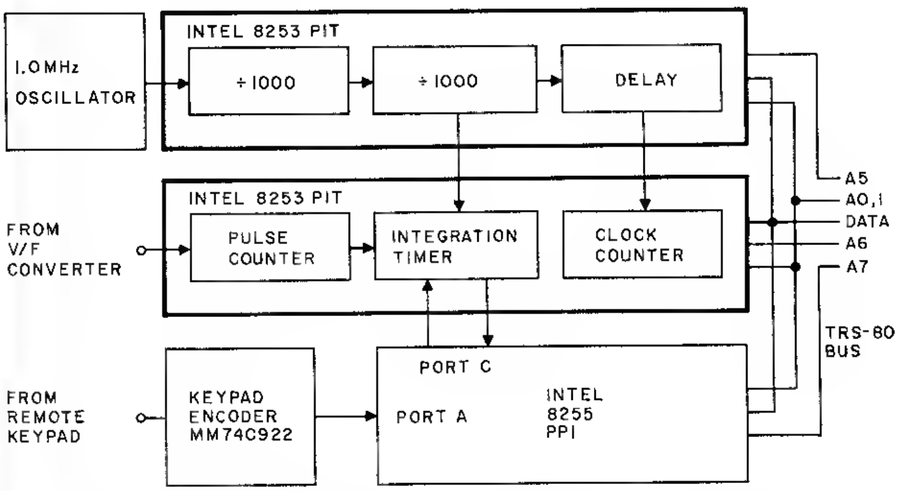


Fig. 2. The complexity of the interface is minimized by using programmable interface chips that connect directly to the TRS-80 bus. The top 8253 is used as a clock divider and software setable delay (1 ms steps), while the second 8253 counts up the pulses from the photometer, controls the integration time (software setable), and keeps track of the time. An 8255 provides parallel I/O, and an encoder and clock oscillator round out the interface.

asked for. The computer tells the operator which star to record and where to set the knobs and levers. Once this is done, the "go button" is pressed and the computer takes full control.

Displays Light Curve

As the TRS-80 records the light from the star, it displays the light curve on the monitor—twinkles and all. A thin, unnoticed cloud drifting by will cause the line on the display to go bananas, which is a hint to the operator that when the TRS-80 asks him what to do with the observation, he should either repeat or terminate, not data ok, continue.

As each observation is made, it is added to a matrix displayed on the monitor. It is easy then to see if, on the last observation, the operator, somewhat sleepy and cold at 2 am, forgot to flip a mirror or change a filter. He has a chance to repeat the observation before a bad one ruins the whole sequence.

At the end of each sequence, the printer is ordered to print the raw data for the record. The data is then reduced by the computer right on the spot and results printed and displayed.

The computer also includes a statistical analysis of the sequence, giving the operator nearly instantaneous feedback about the weather. If the night is bad for photometry, it's nice to know after the first star. One can then hang it up, instead of charging along until the wee hours, only to find the next day, after hours of data reduction, that it really was a hopeless night.

If you have a complex real-time control task eating your lunch that would be better handled by a computer, you ought to consider seriously the TRS-80 and an adaptable interface. Be forewarned, however, that you can quickly become addicted to the efficiency, flexibility, and power of a full computer operating in a high level language. It may become the most important "person" on your staff.■

Photo 3. The "brain" of the Fairborn Observatory is an unattended TRS-80 and small six-chip interface designed by the author. Once the program is loaded, the video monitor is turned off and control is maintained from the telescope, some 60 feet away.

✓See List of Advertisers on page 306

A new level of sophistication for disk-equipped Model Is.

LDOS

Paul Wiener
RFD Box #355
Jaffrey, NH 03452

LDOS
Lobo Drives International
Goleta, CA
$149

First of all, let me relieve the suspense by stating that, in my opinion, LDOS is by far the best disk operating system (DOS) currently available for the Model I TRS-80.

I've owned a TRS-80 disk system for some time and have had the opportunity to play with ULTRADOS, two versions of NEWDOS, two version of VTOS, two of DOS PLUS (including DOUBLE-DOS), and three of TRSDOS. So my statement is based on solid experience.

I'll evaluate the two major aspects of a DOS, its user support (including documentation) and the DOS itself.

Documentation

The current official version of LDOS documentation (Ldocs) is 253 pages which, according to Bill Schroeder of Galactic Software, LDOS project leader, took about one and a half years to write. It's broken down into sections which describe the features and commands of the system for the everyday user, and sections with specialized technical information for the systems level programmer.

Every command has its own section in the documentation and each of these sections has its own sequence of page numbers. The manual lacks an index, and the repetitive numbering scheme may make it difficult to provide one in the future.

The writing style is a little more technical than it needs to be. For instance, tracks are usually referred to as cylinders, and hexadecimal numbers are shown in the old Datapoint format—for example 5C00H is shown as X'5C00'. But no one should be stymied by the language, thanks to the numerous examples. For instance, two pages deal with Purge and include five specific examples showing how it is used with its various parameters, and detailing the results in each case.

Similar explanations are also provided for key LDOS concepts, such as the nature of logical devices, device independence, partial filespecs, and wild-card characters. For example, I learned that a phantom device does not relate to ectoplasmic displays; it refers to a technique that allows you to link two devices which ordinarily can't communicate directly. This is done by creating a third logical device to act as a bridge. I also learned that a device-independent DOS is an operating system that allows executive commands relating to I/O routing to override subsequent I/O routings specified at lower command levels. What this means is that you can do tricks like having LPRINTs go to a disk file or the video, instead of a printer, which is quite handy when you have no printer available.

A sampling of information in the Ldocs technical section includes such goodies as maps of system entry points, (including some in Level II ROM), explanations of the directory, Device Control Blocks (DCB's), File Control Blocks (FCB's), and file formats.

This section also contains comprehensive explanations of the more arcane features of the DOS. A case in point: FILTER is a command that was introduced by VTOS 4.0, but with little information on what to do with it. LDOS also has FILTER. The documentation has over a page of information on how to use it, including several examples. And further, the feature section includes an additional page and a half of commentary, plus five pages of commented, assembled source listings that demonstrate how to write your own filter routine.

Finally, Ldocs also contains a five-page glossary and a six-page error dictionary which goes through each of the 41 LDOS error messages.

Other Aspects of User Support

The LDOS team has taken an interesting approach to the piracy problem. There is no special protection against backing up the LDOS master disk, nor are the Ldocs printed in copyproof turquoise. Instead, LDOS offers extensive customer support, and does its best to make it available only to registered purchasers of LDOS. Here's how:

- The LDOS development team is constantly working on upgrades and patches. Any certified user may send in his/her master disk, plus return postage, and, for no additonal charge, receive the latest version of LDOS by return mail.
- A newsletter containing suggested uses for LDOS features and offering new patches will be mailed quarterly to registered users only.
- Valid users have access to an LDOS bulletin board on Micronet.
- A toll free number exists for the sole purpose of user support, not ordering. If you have questions about LDOS, or are experiencing difficulties, you can call this number and provide the LDOS consultant with your name and valid registration number.

I tested the LDOS customer service number to check on the quality of their assistance. I asked a technical question and was

Star Warrior:
Slay the Dictator
and Save the
Civilized People
STAR WARRIOR
ARMED WITH NUCLEAR MISSILES, BLASTER & POWER GUN, YOU'RE ON A SEARCH & DESTROY MISSION AGAINST A MILITARY DICTATORSHIP!
STAR WARRIOR
EPYX
You owe yourself some fun time and you have just loaded the new Star Warrior into your computer. Suddenly, you are one of the Furies ...and retribution is your business.
The Interstellar Union of Civilized People has been annexed by a soul-crushing dictatorship. You have been given the mission to assassinate the military governor and then destroy the installation. That's Scenario 2.
In Scenario 1, you are on a diversionary mission for the assassination attempt. You want to do as much damage to the enemy units as possible.
The terrain is dangerous and difficult; the enemy forces are powerful and varied. (You may not see the enemy even when you are both in the same general area...but, on the other hand, the enemy may not see you either.) There is always the chance of an attack by an invisible enemy.
There are two scenarios and five levels of difficulty...level five is almost impossible to win. There are usually no fixed playing times for Scenario 2; you play until you get the governor and escape or until you are killed. But you can even set a fixed playing time in Scenario 1.
You have the choice of suits and equipment, and of movement, combat and special commands.
In other words, like all EPYX games, there are enough variations in the game so that you will never tire of playing. Each game is different and fresh. You will never get bored playing the game. Not in your or your computer's lifetime.
Star Warrior. Another bug-free, fantastic game from EPYX. With the unique EPYX lifetime warranty: If anything happens to your cassette or disk at any time and for any reason, send it back with just $5.00 for shipping and handling and we will send you a brand new one.
(Of course, there is also our 30-day unconditional guarantee: if your EPYX game has any defect whatsoever within 30 days of purchase, return it to us or your dealer and we will replace it free. No questions asked.)
Visit your dealer now and pick up Star Warrior in its good-looking, protective box with one of the best instruction books you ever read. Now available on disk for the Apple II (48K RAM with Applesoft) and Radio Shack's TRS-80 (32K RAM). And on cassette for the TRS-80 (Level II, 16K) and the Atari (32K). Only $39.95 on disk or cassette.
If your dealer is out of stock and you can't wait, order directly from Automated Simulations. $39.95 plus $2.00 for shipping and handling (and sales tax if you are in California).
We guarantee you will be delighted with Star Warrior. (If it is not exactly what you want, return it to us for full refund. Again, no questions asked.)
Enclose your check. Or if you order by Visa or MasterCard, use our toll-free phones: In the United States: operator 861 (800) 824-7888; In California: operator 861 (800) 852-7777; In Hawaii and Alaska: operator 861 (800) 824-7919.
Order today. You and your computer deserve the fun.
AUTOMATED SIMULATIONS, INC.
Dept. SW3. P.O. Box 4247
1988 Leghorn Street
Mountain View, CA 94040
48
EPYX
COMPUTER GAMES THINKERS PLAY

soon speaking to one of the members of the LDOS development team. Aside from knowing his subject thoroughly, he had good communication skills and did not try to make me feel like an idiot for knowing less than he. I asked what their reaction would be if I were to unearth a major bug in their system. He told me that if anyone reported a repeatable failure of LDOS to perform as documented, the problem would be addressed immediately and a remedial patch developed within days. Any users who had reported the problem would be called back and given the fix. Others would be informed through the newsletter. Of course, the error would be zapped for all future releases.

DOS Features

The features of LDOS are too numerous to relate in-depth. Instead, I will touch upon a number of items which I especially like, and then move on to some of the major factors which I feel make LDOS really outstanding.

LDOS offers various changes and improvements to the TRS-80 keyboard response. Debounce is, of course, one of them. So is auto-repeat. A key-stroke multiply (KSM) package allows the user to custom define the keys. In this connection, the clear key is used as a special control key. Shift, down arrow functions as a standard control key. If your system hangs up during disk operation because you forgot to insert a disk in drive 0, or close the drive door, etc.,LDOS allows you to correct the situation by pressing shift Break to restart the drive, thus allowing I/O to resume.

During a system boot, depressing clear, Enter or up arrow, respectively, averts automatic system reconfiguration, AUTO commands, and key debounce with auto-repeat. Holding down D boots the system directly into Debug.

System reconfiguration refers to the SYSTEM command and its various parameters. This command allows you to alter certain aspects of the system's intrinsic performance. For example, you can tell the DOS that you only have one disk drive. Then, when you've typed the name of a nonexistent file, you'll no longer have to wait while the system attempts to look for it on three equally nonexistent drives. Even if you're unlucky enough to have more than one drive, you can still use System to tell LDOS what kind each is (hard, soft, 8 inch, 5¼ inch), how many tracks each has, and how fast a step rate each will support. You may configure a different type of drive for each slot. If you happen to have a Lobo expansion interface, you can have up to eight drives on line at once!

Another System command is ALIVE. This keeps a graphics block wriggling at the top-right corner of the screen whenever the interrupt-task processor is active.

Other System parameters allow you to tell your computer what you want the cursor to look like and whether or not it should blink, invoke lowercase display (if you have a lowercase modification), and tell the system whether you want a screen print option.

System also lets you select the type-ahead option. Type-ahead allows you to go on typing while the computer is busy performing other functions.

Serious applications programmers may be interested to know that LDOS can pre-allocate file space via the CREATE command, offers a file mode which updates the EOF after each write, and provides a means to change the logical record length of a file. Also, new file modes exist in BASIC which allow you to open a file only if it is new, or only if it is old. Thus, you can protect your programs from meddling with information that was meant to be archival, or conversely, prevent the creation of a new file when an old one was meant to be updated.

LDOS provides a number of enhancements to BASIC (BASIC must be copied over from a TRSDOS disk). These include the new file modes alluded to above, blocked (variable length) records, a program single stepper, and several new CMD "n" statements. CMD's 0, P, N, and X, respectively, turn off Breaks ability to send you to Debug, for a screenprint, renumber a program, and provide a variable cross reference utility.

LDOS also provides an extended debugging package that goes beyond the capabilities of TRSDOS's Debug. It may be used on code either in memory or on disk. Yet, it is

"Device independence is an important mainframe feature which this DOS brings to the TRS-80."

not quite as handy for disk editing as one of the ZAP family monitors because the user has to load the information off disk, alter it in memory, and then write it back to disk.

There is also a sort of mini-monitor available right from the DOS, via the Memory command. LDOS honors HIGH$, which means that it protects programs in high memory, as does BASIC. Memory tells you the current HIGH$ (Memsize reserved) or, you can use it to set HIGH$ to a new value. You can also use Memory to inspect any word in memory or to alter any byte or word in RAM. And you can use Memory to jump directly from DOS to any specified address. I think this mini-monitor is one of LDOS's fine touches.

Major Features

LDOS offers a surprisingly comprehensive Job Control Language (JCL), extended device independence, and a wide spread acceptance of partial filespecs (with wildcard characters).

The way the system handles defaults for file name extensions is an excellent example of how LDOS was designed to support sophisticated users while making things as simple as possible for the beginner.

Consider a case familiar to all TRSDOS veterans. I have a machine code business applications package saved on my disk under the file name INVADERS/CMD (all right, so it's not a business program). If I want to run it, I have to enter its name, but I don't have to type the /CMD at the end. The system fills that in for me as a default extension. Likewise for DUMP and /CIM.

LDOS provides many automatic default extensions not found in other DOSs. Suppose I enter the command LIST REVIEW. Because /TXT is the default extension for LIST, the system searches for a file called REVIEW/TXT. However, if such a file is not located, the system doesn't return with a File Not In Directory message. Instead it does a second search, this time for a file called Review.

Another aspect of the LDOS design philosophy is manifested in its upward compatibility with TRSDOS. It does not promise to mesh perfectly with NEWDOS files, but its PROT command does attempt to render alien disks readable by LDOS without diminishing their readability by the other system.

Device independence (DI) is an important mainframe feature which this DOS brings to the TRS-80. Some of you might be familiar with VTOS's DI. LDOS, like VTOS, allows routing, linking, copying, filtering, appending, setting and resetting of devices/files, as well as the creation of your own new logical devices. Unlike VTOS, LDOS provides adequate documentation on how to exploit these capabilities.

Still another feature that has been expanded by LDOS is Job Control Language. For those of you not familiar with JCL, it's a sort of high-level language that lets you give preprogrammed commands to the DOS.

The earliest versions of TRS-80 JCL were limited to a simple concatenation of direct DOS commands. For example, on power up you could automatically turn on Verify, the Clock and Trace, then go into BASIC, set an appropriate memory size, file number, and load and run a BASIC program.

VTOS 4.0 made some impressive additions to JCL, such as conditional execution (much like BASIC's IF...THEN...GOTO statement), keyboard input during a job (like BASIC's INKEY$ statement), nested JCL files) analogous to chained BASIC programs), and token replacement (something like BASIC's variables).

LDOS retains these features and has added procedure labeling. Labels may be used in a JCL file to create a space saving procedure-library (Proc-Lib). The Flash micro has been altered to flash a single line instead of the whole video screen, and Alert has been changed to put variable audio tones out the cassette port instead of wearing out the relay. A new input macro is similar to BASIC's input statement.

JCL may seem complicated at first, but the manual has several thorough examples. By the way, LDOS has enhanced the utility of the BUILD command, normally used to create JCL files, by introducing hex and append parameters. The former allows the input of values not normally possible through the keyboard. The latter facilitates extending old files without a total rewrite.

But what about that partspec and wildcard stuff? Well, supose that you had given all your Scripsit files names with /SCR extensions, and that you wanted to know what Scripsit files were present on your disks. You could enter DIR /SCR and LDOS would respond with a list of all the Scripsit files on the disks currently on-line. /SCR is an example of a partspec. Now for that wildcard function: "$" (dollar sign) is the wildcard character. Suppose you had the following four files on a disk: Cat, Cot, Pat and Car. Entering DIR $AT would return Cat and Pat. DIR C$T would return Cot and Cat. DIR CA$ would return Cat and Car. The use of partspecs and WCC's is not limited to DIR. LDOS commands accept them wherever the system designers thought it useful.

Whenever a disk file is created or updated, the date is included in its directory entry. It may be displayed by the DIR (A) command. Also, files which have not been backed up since their creation or last update are flagged in the directory by placing a plus sign (+) next to their name. You can use a class backup procedure to copy all such flagged files with a single command.

✓See List of Advertisers on page 306

"Now that I've extolled its virtues, I'd like to devote some space. . . to some of LDOS's drawbacks."

There are three kinds of backup available in LDOS: mirror image backup, backup by class and backup reconstruct. Mirror image backup refers to the familiar one-on-one backup. A form of backup by class was exemplified in the preceding paragraph. Backup reconstruct applies to backups from one type or size disk to another.

The XFER command, and the (X) parameter for the copy, backup, load and run commands, make it possible for one drive owners to use and manipulate material on non-LDOS disks. You can even use RUN (X) to run machine language programs on non-system disks, as long as they're formatted.

PATCH is a command which some DOS's offer to modify existing software. Once a patch has been created, it is easier to use than the ZAP method of code replacement. LDOS allows patches to be created in either of two modes: the D (for Direct) mode is disk space efficient but irrevocable. The X (or hex) mode uses more disk space, but can be undone at any time via a Yank.

LDOS comes with the best Spooler I've yet seen for the TRS-80 Model I. Spooling allows your computer to go about its other business while feeding information to a printer which can't keep up with it. This is done by buffering the printer-bound output and sending it a little at a time during interrupt processing. LDOS's Spooler allows you to specify whether the buffer is to be in memory, on disk, or both.

Also present on the LDOS disk is a utility called CMDFILE. This is somewhat like Tapedisk or LMOFFSET, except that it goes farther than either. CMDFILE allows transferring object files from tape to disk and vice versa. It allows the merging of several object files into one load module. Files may be offset before being resaved, with an optional relocating appendage. You can offset only a section of a file, if part of that file loads to video memory or the system vector page of RAM. You may specify the status of interrupts and keyboard debounce in the file to be dumped. When CMDFILE reads in a file it displays the memory block(s) occupied by that file as well as the file's transfer address. This, as well as the entire CMDFILE dialogue, can be logged to a printer.

LDOS boasts an extremely powerful terminal program called LCOMM. In combination with LDOS's RS232 drivers, it allows reliable telecommunications at up to 300 baud without handshake. LCOMM automatically uses all available memory to buffer incoming data. LCOMM offers an on-screen menu, which can be called when needed, and supports full and half duplex modes—with or without echo. LCOMM provides for ASCII disk file transference, and has several other features. My only complaints about this utility are: ASCII files are the only kind which may be sent/received, and when disk files are being transferred, the user is responsible for opening and closing the file, and positioning the file to its beginning or end. Failure to remember some of the details could result in information being lost.

For all you word processing buffs, LDOS comes with patches to make Scripsit and the Electric Pencil LDOS compatible. These patches also contain enhancements. Patched Pencil will honor HIGH$, leaving high memory intact. It will also work with LDOS's keystroke multiply feature. Patched SCRIPSIT, which also honors HIGH$, allows you to view disk directories. It will also use whatever driver is pointed to in the printer DCB.

Drawbacks

Now that I've extolled its virtues, I'd like to devote some space to what I think are some of LDOS's drawbacks.

One feature which may not be popular is the absence of Disk BASIC from the LDOS master disk. The user must copy BASIC over from a TRSDOS 2.2 or 2.3 disk. However, once that has been attended to, LDOS patches BASIC and offers several enhancements.

Other features available on competing DOSs but not offered in the current release of LDOS include ULTRADOS's selection of three BASICs with three different trade-offs of features versus memory and NEWDOS–80's MINIDOS. LDOS also lacks NEWDOS–80's new file modes, but it does have a few new and extremely useful ones. At this time, LDOS does not support the Percom double density board, but plans to in the future. LDOS does offer immediate double density to those who have the Lobo expansion interface.

The most serious problem I've come across is the possibility of losing data if you kill a file while it's still open. The consequences of killing an open file in LDOS are nowhere near as dire as they were under early releases of TRSDOS. I have reported my findings to LDOS customer service and corrections are being made. In the meantime, users will be cautioned in the newsletter. A similar problem is that innocent files may be overwritten if a disk fills up while the printer is routed to a file. A warning exists in the Ldocs, that tells you to make sure you have sufficient disk space before engaging such a routing.

On the whole, I've found LDOS to be as error-free as any DOS I've ever worked with. In human engineering, system-integration, and flexibility it runs cylinders around the others. I think it's well worth its price.■

✓See List of Advertisers on page 306

A plug-in Z-80A black box to get you running at 4 MHz.

Plug Compatible Processor

Paul Vonk
P.O. Box 271
Dillard, GA 30537

I wanted to speed up my TRS-80 as soon as I got my Model I home and started playing backgammon. Explaining to friends why a game of backgammon took longer on a "high speed computer" than playing the old fashioned way was embarrassing.

A glance at the schematic[1] revealed that the clock frequency on the Z-80 CPU chip was 1.7 MHz. I checked into the various clock mod kits available.[2,3,4] They were certainly cheap enough, $25 to $40, but would they do the job? The first thing I noticed is that I would have to open the keyboard to install the clock mod. Of course, this would void the warranty. No problem, after the 90 day warranty period was up.

To install the clock mods, one must solder several wires, cut one or more traces and even drill a few holes in the cabinet. I began to wonder how much damage a butcher like me could do. Also, I had heard rumors about a marginal power supply in the Model I, and the technical manual warns against adding more of a load. I wondered how an already hot transistor voltage regulator would respond to the increased load of the extra components in the clock mod.

I could see no reason for increasing speed if it would be unreliable. There seemed to be little point in running a program twice as fast if the result might be wrong. Most of the clock mods don't guarantee 4 MHz, and some of the instructions for it seem formidable. Also, you may have to buy more parts.

A return trip to the schematic revealed some of the problems with trying to operate at 4 MHz. The Z-80 in the Model I is only guaranteed to 2.5 MHz. The solution is easy enough—buy a Z-80A for $15, which is guaranteed to 4 MHz. The second problem isn't so easy to solve. The memory timing circuit violates several timing specs on the RAM if the clock is simply sped up. Plugging in faster RAM won't help—the problem is in the circuitry which generates the CAS, RAS, and MUX signals. This means that memory read/write errors can occur. I began to despair. I needed something reliable that I could just plug in.

The MicroCompatible Inc. Plug Compatible Processor (PCP)[5] is just what the name implies—a plug compatible (it plugs into the keyboard expansion port) processor (a complete Z-80A processor and support circuitry). The 13 × 2 × 3 inch PCP sits on the desk behind the keyboard. A 40-pin connector plugs into the keyboard expansion port. The cable to the expansion interface plugs into the top of the PCP. All that remains is to plug the power cord into the wall.

The PCP contains a 4 MHz Z-80A CPU. But merely speeding up the Z-80A to 4 MHz will exceed the timing specifications on the 4116-2 RAM chips. Support circuitry in the PCP generates special RAS, CAS, and MUX signals which allow reliable 4 MHz operation without inserting wait states to allow the CPU to wait on the RAM. In fact, after simply changing a jumper plug inside the PCP, I've been running reliably at 4.6 MHz.

Several additional problems could arise with 4 MHz operation. The Model I uses software timing loops to generate certain timing signals. In order to read a tape which was written at normal speed, the processor must again be slowed to the original speed. Of course, a tape may be written and read at high speed (and at twice the baud rate).

Floppies will not work at high speed with TRSDOS. When the TRS-80 is running at high speed, the control signals timed by the software loops are no longer compatible with the Floppy drives. The software keyboard debounce routine also uses a timing loop. At high speed the time in the loop is more than cut in half. If the processor is not slowed down during keyboard accesses, some keybounce may result. The PCP automatically returns to 1.77 MHz during disk accesses and keystrokes to avoid these problems. A high speed disk switch is provided for NEWDOS 80 users.

When you first get your PCP, you'll notice two more jumper plugs which must be set to configure the PCP for your particular system. Radio Shack has continually had problems driving the RAS, CAS, and MUX lines to the expansion interface. If these signals do not arrive at the RAM chips with the correct

timing, RAM read/write errors will result. Radio Shack has tried many fixes, and these modifications necessitate setting the jumper plugs properly.

As a final coup de grace to the RAM error problem, Radio Shack did a total redesign of the expansion interface board somewhere between the boards numbered 200,000 and 220,000. The new expansion interfaces receive the RAS signal from the keyboard (or PCP) and generate the MUX and CAS signals on the same card as the RAM. Unfortunately, the processor must slow down to normal speed during RAM accesses to the top 32K of memory in the expansion interface.

However, all's well, or nearly well, in the end. This high order memory access slowdown doesn't really cost too much time because most of the memory accesses are to the BASIC ROMs and not to the top 32K of RAM. The last jumper in the PCP allows the high order RAM access slowdown to be switched off for optimal operation with the older expansion interfaces.

Owners of early Model Is which haven't been modified receive an additional benefit. The PCP allows the use of the expansion interface. Early Model Is had a +5 volt lead brought out to the keyboard edge connector. This same pin was conveniently grounded in the expansion interface. Plug the early Model Is into an expansion interface and ZAP!, no more +5. This is why the first page of the Expansion Interface Manual[6] states: "Note: Don't connect the Expansion Interface to a Level 1 TRS-80. The two are not compatible." Thank you, Radio Shack. ■

1. TRS-80 Micro Computer Technical Reference Handbook. First edition. Radio Shack catalog no. 26-2103. pp. 107-108.
2. Mumford Micro Systems, Box 435-E, Summerland, CA 93067.
3. Simutek, P.O. Box 13687-z, Tucson, AZ 85732.
4. Archbold Electronics, 10708 Segovia Way, Rancho Cordova, CA 95670.
5. Microcompatible Inc. P.O. Box 107, Scaly Mountain, NC 28775.
6. Expansion Interface. Radio Shack catalog no. 26-1140/1141/1142.

Games from BIG FIVE will turn your computer into a

TRS-80 HOME ARCADE

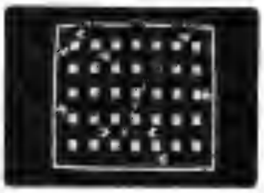
SUPER NOVA©

If you and your TRS-80 have longed for a fast-paced arcade-type game that is truly a challenge, then **SUPER NOVA** is what you've been waiting for. In this two player machine-language game, large asteroids float ominously around the screen. Suddenly your ship appears and you must destroy the asteroids before they destroy you! (But watch out because big asteroids break apart into little ones.) The controls that your ship will respond to are thrust, rotate, hyperspace, and fire. All right! You've done it! You've cleared away all the asteroids! But what is that saucer with the laser doing? Quick! You must destroy him fast because that guy's accurate!

GALAXY INVASION©

The sound of the klaxon is calling you! Cruel and crafty invaders have been spotted in battle formation warping toward Earth at an incredible speed. Suddenly, your ship materializes just below the huge flock of invaders. Quickly and skillfully you shift right and left as you carefully fire your lasers at them. But watch out! A few are breaking out of the convoy and flying straight at you! As the whine of their engines gets louder, you place your finger on the fire button knowing all too well that this shot must connect—or your mission will be permanently over! With sound effects!

ATTACK FORCE©



Your TRS-80 screen has been transformed into a maze-like playfield for this game. As your ship appears on the bottom of the screen, eight alien ramships appear on the top. All of them are traveling at flank speed directly at you! Quickly and boldly you move toward them and fire missiles to destroy them. But the more aliens you destroy, the faster the remaining ones become. If you get too good you must endure the wrath of the keeper of the mazefield: the menacing "Flagship". You must destroy him fast because, as you will find out, that guy's accurate! With sound effects!

COSMIC FIGHTER©

With thousands of stars whizzing by you, your **SPACE DESTROYER** ship comes out of hyperspace directly under a convoy of aliens. Almost effortlessly, you skillfully destroy every last one. But before you can congratulate yourself, another set appears. These seem to be slightly more intelligent than the first set. Quickly you eliminate all of them, too. But your fuel supply is rapidly diminishing. You must still destroy two more sets before you can dock with your space station. All right! The space station is now on your scanners! Oh no! Intruders have overtaken the station! You must skillfully fire your neutron lasers to eliminate the intruders from the station before your engines run out of fuel and explode! With sound!

METEOR MISSION II©

As you look down on your space viewer you can see the stranded astronauts that are crying out for you to rescue them. But first you must maneuver your shuttle down through the asteroids & meteors before you can reach them. Great! You've got one! But now can you get back to the space station to save your fellow shipmate or will you crash and kill both of you? You can fire your lasers to destroy the asteroids, but watch out, because there could be an alien FLAGSHIP lurking behind! Includes sound effects!

NEW JOYSTICKS!

For $39.95 it's now possible to have the famous ATARI joystick interfaced with your Model 1. All of our tapes are now completely compatible with the joystick. Packaged with complete instructions, you can even use it with your own programs! If your old tapes do not say "Joystick Version" on them and you wish to exchange them for new Joystick versions, enclose $2 and your old tape. (Call or write for info on Mod 3 joysticks.)

BIG FIVE SOFTWARE

✓357

P.O. Box 9078-185, Van Nuys, California 91409

For MC/Visa or COD orders, call (213) 782-6861 from 9am-5pm Pacific time.

Prices per game: Level 2, 16K Mod 1/Mod 3—$15.95
Level 2, 16K Mod 1/Disk—$17.95
10% discount for 2 items, 15% for 3 or more. (Including Joystick)
Please add $1.50 postage/handling, Calif. residents add 6% tax.
All games are written in machine language and supplied on cassette.
Disk versions save high scores to your TRSDOS or NEWDOS diskette.

Cassette versions require 16K memory, disk versions require 32K.
We accept checks, money orders, and MC/Visa orders.
All games ©1980 by Bill Hogue & Jeff Konyu.
TRS-80 & TRSDOS are trademarks of Tandy Corp.
NEWDOS is a trademark of Apparat, Inc.
Dealer inquiries invited.

Give your TRS-80 the gift of sight.

EYE-80

James S. Hawkes, PhD
25 Bainbridge Dr.
Charlestown, SC 29407

Grady R. Reese
500 Tulip St.
Summerville, SC 29483

It's exciting when your computer accomplishes a task that you couldn't or wouldn't do yourself. Such an occasion recently came up when we were investigating a problem of automated data collection.

The Problem

During the spring semester the College of Charleston offers a course in Operations Research which covers topics in queueing theory and simulation. To apply the queueing theory and simulation modeling techniques, we hoped to gather the actual arrival times of customers entering a bank for service. However, to obtain meaningful results it would be necessary to collect this data for a prolonged period—perhaps several weeks. The thought of asking class members to spend several weeks collecting data seemed impractical, and the idea was nearly scrapped.

Fortunately, we had a TRS-80 handy. Surely the TRS-80 wouldn't mind sitting in a business for several weeks and probably wouldn't complain about being cooped up in a box, particularly if the box contained a fan.

Using the photo cell interface described here will allow the TRS-80 to detect an object and to some extent determine its speed and direction. This interface opens the door for many applications including security, energy management, and the data collection system described below.

Date Collection Application

A bank is a perfect environment is which to apply queueing and simulation models if sufficient data can be collected. Anyone who has been in a bank knows that there are lines and servers, and that you have to wait in order to be served. You probably wondered why that bank didn't have more tellers. It would certainly be pleasant never to wait in a line at a bank. However, this would require an enormous number of employees, particularly during busy periods, and thus would be cost prohibitive.

To find the optimal number of tellers one must strike a balance between the bank's cost of serving and the customer's cost of waiting. Since the latter cost is nearly impossible to obtain, it is difficult to determine the optimal number of tellers needed by a bank. Simulation and queueing models can provide useful information about system performance under various staffing levels. The first bank we called agreed to help.

The bank we selected used a single line to feed all the tellers. This made our job even easier. We could place the device at the beginning of the line and record customer arrivals and balks (leaving the line after entering) as they occurred. These were

LDR1	1 EACH	PHOTOCELL	RS #276-116
VDR1	1 EACH	100 kilohm POT	RS #271-220
R1	1 EACH	2.2 kilohm RESISTOR	RS #271-027
Q1	1 EACH	PNP TRANSISTOR	MOTOROLA HEP 50032
MISC			
	1 EACH	PCB OR PERF BOARD	RS#276-1394
	1 EACH	DIP CABLE	RS #276-1976
	1 EACH	16 PIN IC SOCKET	RS #276-1998
	1 EACH	5 VOLT REGULATED	POWER SUPPLY

Table 1. Parts List

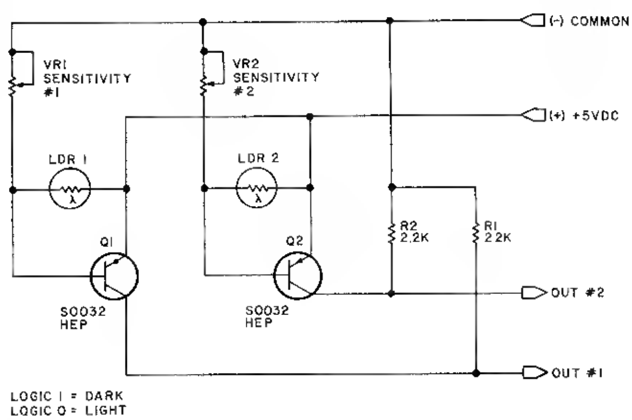


Fig. 1. Schematic for EYE-80

written on disk, and at the end of the day, one of the employees would collect the disk and turn off the TRS-80.

In addition to collecting the arrival and balk times, we also recorded the elapsed time between the two photocells. From this information the speed of the object passing through the cells could be determined. This information was quite useful in detecting false entries and balks which could be caused by swinging arms.

In total, nearly 10,000 pieces of data were collected during the three weeks. This data was transmitted to the school's minicomputer where it was examined and false entries and balks were removed. The students analyzed the data then estimated the parameters of queueing models. In addition, they developed a simulation model of the bank's teller system in which they not only provided for different system configurations but also provided a graphic simulation of the bank's operation. The graphic simulation was particularly interesting since the manager could schedule tellers and then watch the performance of the proposed schedule.

Hardware

Our data collection device is called EYE-80, the heart of which is the photocell.

The photocell by definition is a light-dependent resistor. This means that the resistance of the device will change depending on the intensity of the light that falls on its sensing surface. In our design we used RS part #276-116 which will change its resistance from about 100 Ohms in bright light to five megaohms in total darkness. The variation in resistance is used to control the conduction of a transistor switch. Therefore, the eyesight suggested by the project name is blurred at best, as it can detect only the presence or absence of light.

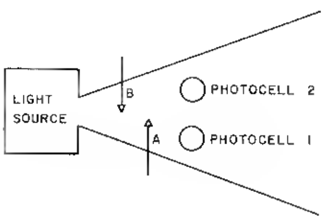


Fig. 2.

Refer to the schematic diagram (Fig. 1) as we discuss how the characteristics of the photocells are put to use. Notice that the diagram contains two identical circuits. The parts for one complete detector will be designated by 1s, and the second detector by 2s.

As indicated in Fig. 1, the photocell for circuit 1 (LDR1) is placed across the emitter and base of Q1. The operation of a PNP transistor requires a potential difference between the emitter and base with the base more negative than the emitter. This condition will result from our configuration when the photocell is darkened, thus presenting little impedance to normal emitter base current flow. This causes the transistor to be turned on, and consequently a +5 volt (logic 1) is presented at output #1.

Now let's see what happens when the photocell is well lit. LDR1 will now present a low impedance to the emitter base junction (100 Ohms). This shunts the bias for the transistor through the photocell causing the transistor to turn off. Resistor R1 is used as a pull-down resistor so that a logic 0 is now presented at output #1. Resistor VR1 is used to set the bias level for Q1, and therefore its sensitivity to light changes. The other half of the circuit is identical in composition and operation.

Interface Connections

The connection of the EYE-80 is simple and straightforward. All that is necessary is to connect the output #1 and output #2 bits from the EYE-80 to two bits of the input port you choose. We used the PPI-80 parallel port as described in the Sept. 1980 issue of *80 Microcomputing*. After providing a common ground connection from the host computer to the EYE-80 and to all unused input port bits, you must provide a +5 volt regulated source.

Since EYE-80 only produces two information bits, it can be interfaced to the TRS-80 through the PPI-80 parallel port or it could be attached, with the appropriate cabling, to any parallel port.

If the PPI-80 is used, one of the 8255's ports must be configured as an input port. This can be done quite easily by sending the appropriate control word to the 8255's control port. See line 1 of the BASIC listing (Program Listing 1). Once the port is properly configured, receiving information from the port is easy. We will discuss how this can be done in the context of our application.

Software Considerations

Since our application required not only the direction, but also the speed of each entering customer, we felt it was necessary to use assembly language to monitor the photocells. However, since we needed to write the collected data to disk, we decided to write part of the program in BASIC. The programs were linked by the USR ( ) function.

As discussed earlier, EYE-80 generates a +5 voltage if the photocell is shadowed. This voltage is the standard voltage associated with a binary 1. Therefore, to detect the shadowing of a photocell, look for a positive number on the port being scanned. Receiving a positive

Program Listing 1. EYE-80 BASIC Listing

```
1 OUT 3,128 : 'OPENS PPI-80 PORT FOR INPUT
5 DEFUSR1=&HE99E
10 DEFINT T,A,H,L
15 'INITIALIZE T,T1,T2 TO 0 WHERE :
16 'T - HAS VALUE OF 1 IF PERSON IS ENTERING
17 '        VALUE OF 2 IF PERSON IS EXITING
18 'T1 - NUMBER OF FULL COUNTS OF 256 REACHED BY TIMER
    BETWEEN AN ENTRANCE OR AN EXIT
19 'T2 - PARTIAL COUNT OF 256 REACHED BY TIMER
20 T=0:T1=0:T2=0
30 A=VARPTR(T)
40 GOSUB 1000
45 'NOW POKE ADDRESS OF T INTO ASSEMBLER ROUTINE AT VAR
    CG:
50 POKE -5736,LB
60 POKE -5735,HB
70 A=VARPTR(T1)
80 GOSUB 1000
85 'NOW POKE ADDRESS OF T1 INTO ASSEMBLER ROUTINE AT VA
    RT1:
90 POKE -5734,LB
100 POKE -5733,HB
110 A=VARPTR(T2)
120 GOSUB 1000
121 'NOW POKE ADDRESS OF T2 INTO ASSEMBLER ROUTINE AT V
    ART2:
122 POKE -5732,LB:POKE -5731,HB
130 H=USR1(0) :'JUMP TO MACHINE LANGUAGE PROGRAM
140 PRINT T,T1,T2
150 GOTO 130
1000 HB=INT(A/256)
1010 LB=A-(HB*256)
1020 RETURN
```

Program Listing 2. EYE-80 Assembler Listing

```
E998            00100           ORG     59800
E998 0000       00200 VARCG:    DEFW    0         ;COMING OR GOING VARIABLE
E99A 0000       00300 VART1:    DEFW    0         ;TIME 1 VARIABLE
E99C 0000       00400 VART2:    DEFW    0         ;TIME 2 VARIABLE
                00500                             ;WAIT UNTIL BOTH BEAMS ARE CLEAR
E99E DB01       00600 CLEAR:    IN      A,(1)     ;GET BYTE FROM PORT
E9A0 F600       00700           OR      0         ;CHECK TO SEE IF EITHER BEAM
                00800                             ;HAS BEEN BROKEN
E9A2 20FA       00900           JR      NZ,CLEAR;CONTINUE UNTIL BOTH BEAMS CLEAR
```

Program continues

```
                 Ø1ØØØ                          ;SCAN PORT FOR BROKEN BEAM
E9A4 DBØ1        Ø11ØØ SCAN:    IN      A,(1)   ;GET BYTE FROM PORT
E9A6 F6ØØ        Ø12ØØ          OR      Ø       ;CHECK FOR BROKEN BEAM
E9A8 28FA        Ø13ØØ          JR      Z,SCAN  ;IF ZFLAG SET THEN CONTINUE
                 Ø14ØØ                          ;TEST FOR ENTRY OR EXIT
E9AA CB47        Ø15ØØ          BIT     Ø,A     ;IF BEAM Ø IS NOT BROKEN Z FLAG
                 Ø16ØØ                          ;IS SET
E9AC 28ØC        Ø17ØØ          JR      Z,PEXIT ;IF FLAG IS SET GO TO EXIT SUB
                 Ø18ØØ                          ;OTHERWISE MUST BE AN ENTRY
E9AE CDD1E9      Ø19ØØ PENTER:  CALL    CLOCK
E9B1 2A98E9      Ø2ØØØ          LD      HL,(VARCG);PUT 1 IN CG VARIABLE
E9B4 36Ø1        Ø21ØØ          LD      (HL),1
E9B6 CDC6E9      Ø22ØØ          CALL    LOADT   ;WRITES TIME VARIABLES
E9B9 C9          Ø23ØØ          RET
                 Ø24ØØ                          ;COLLECT DATA FOR EXITS
E9BA CDD1E9      Ø25ØØ PEXIT:   CALL    CLOCK
E9BD 2A98E9      Ø26ØØ          LD      HL,(VARCG);PUT 2 IN CG VARIABLE
E9CØ 36Ø2        Ø27ØØ          LD      (HL),2
E9C2 CDC6E9      Ø28ØØ          CALL    LOADT
E9C5 C9          Ø29ØØ          RET
E9C6 2A9CE9      Ø3ØØØ LOADT:   LD      HL,(VART2);LOAD LOCATION OF V2
E9C9 7Ø          Ø31ØØ          LD      (HL),B  ;WRITES B TO V2
E9CA 2A9AE9      Ø32ØØ          LD      HL,(VART1);LOAD LOCATION OF V1
E9CD 73          Ø33ØØ          LD      (HL),E
E9CE 23          Ø34ØØ          INC     HL
E9CF 72          Ø35ØØ          LD      (HL),D
E9DØ C9          Ø36ØØ          RET
E9D1 F3          Ø37ØØ CLOCK:   DI              ;DISABLE INTERRUPTS
E9D2 11ØØØØ      Ø38ØØ          LD      DE,Ø    ;INITIALIZE DE
E9D5 Ø6ØØ        Ø39ØØ          LD      B,Ø     ;INITIALIZE B
E9D7 CDE3E9      Ø4ØØØ TIMER:   CALL    LOOK    ;READ PORT
E9DA 28Ø5        Ø41ØØ          JR      Z,RENTR ;IF BOTH CELLS ARE ON THEN RETURN
E9DC 1ØF9        Ø42ØØ          DJNZ    TIMER
E9DE 13          Ø43ØØ          INC     DE      ;ADD 1 TO PRIMARY COUNTER
E9DF 18F6        Ø44ØØ          JR      TIMER   ;CONTINUE LOOKING
E9E1 FB          Ø45ØØ RENTR:   EI              ;ENABLE INTERRUPTS
E9E2 C9          Ø46ØØ          RET
E9E3 DBØ1        Ø47ØØ LOOK:    IN      A,(1)   ;READ PORT
E9E5 FEØ3        Ø48ØØ          CP      3       ;TEST IF BOTH CELLS ARE ON
E9E7 C9          Ø49ØØ          RET
ØØØØ             Ø5ØØØ          END
ØØØØØ TOTAL ERRORS
```

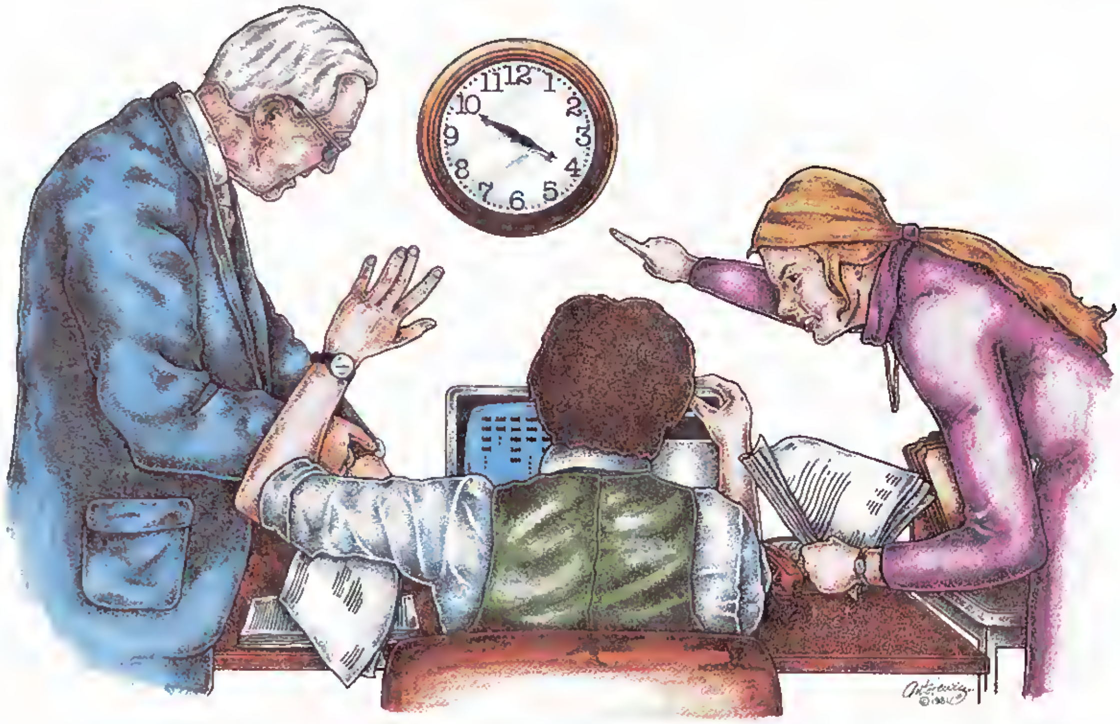
number implies one of the bits on the port has been pulled high by the photocell. All that remains is to determine which cell has been activated.

Sensing Direction

Two photocells can be used to sense the direction of an object moving in a line parallel to the cells. As soon as one of the photocells has been activated (shadowed), the computer must recognize which cell produced the signal. If the photocells occupy bit 0 and bit 1 on the port, then the problem is to decide whether the bit pattern on the port is 00000001 or 00000010. For example, see lines 900 through 1500 in the assembler program (Program Listing 2).

By determining the activated photocell, it is possible to determine the direction of an object passing between the cells. If an object shadows photocell 1 and then shadows photocell 2, it is going in direction A. On the other hand, if photocell 2 is shadowed followed by photocell 1, the object is moving in direction

Four New Programs for May To Keep You Busy Till June!



"Deadlines, deadlines! Don't they realize I'm on a collision course with what must be the biggest ASTEROID in the galaxy?! 'Statistics,' they say. That's what we'll all BE if I can't fend off all these SPACE INVADERS! They're clamoring for reports when it's up to me to intercept enemy Gnat fighters, who are determined to strangle our deep space trade routes! With BALL TURRET GUNNER, any moment could be my last! And if that's not enough, I'm surrounded by Quelon Fighter escorts in COSMIC PATROL . . . and the Quelons never miss! Reports! Statistics! Deadlines! I guess some people just have no sense of priority."

Cosmic Patrol

The Cosmic Patrol program puts you in command of a small interstellar patrol craft. Your mission is to defend Terran space and prey on the Quelon supply ships. Beware of the I-Fighter Escorts—they're armed, fast, and manned by intelligent robots linked to battle computers. . . they NEVER miss! With its fast, real-time action, impressive sound option and superb graphics, COSMIC PATROL is the best program of its kind.

Order No. 0223R tape $14.95 Requires Level II 16K
0224RD disk $19.95 Requires Level II 16K

Asteroid

It's up to you to destroy any asteroids in your sector and to prevent alien spacecraft from infiltrating the Terran Defense Network. Your ship is equipped with an anti-matter cannon. You can shoot large asteroids, but this will shatter them into smaller pieces, each capable of destroying your ship. In addition, alien ships can make instantaneous hyperspace jumps into your area to fire on your ship. You'll need nerves of steel to survive ASTEROID. . . we have no need for non-survivors.

Order No. 0237R tape $14.95 Requires Level II 16K
Order No. 0247RD disk $19.95 Requires Level II 16K

Invaders

The INVADERS are coming! Earth's defenses are all dead except for your laser base. Your assignment is to destroy the approaching invaders. Before Earth's sensors failed, they detected 550 armed invaders in space, speeding toward us in 10 attack formations of 55 each. The sensors detected four different types of attack crafts—large, medium, small and a short profile craft which is the most difficult to destroy. If you can't stop the attacks, Earth is doomed!!

Order No. 0240R tape $9.95 Requires Level II 4K

Ball Turret Gunner

Imagine yourself at the control console of the LW 1417 Stratoblazer, ready for the onslaught of enemy GNAT fighter planes. Your Hindsight Director informs you that a GNAT fighter is closing fast. With your finger tensed over the firing key, you'll have only seconds to react. Suddenly, you see the fire command. With one beam, the Gnat fighter is reduced to an expanding ball of ionized gas. Mission accomplished. BALL TURRET GUNNER with your choice of multiple level of difficulty, optional sound effects and superb graphics, is more than a game—it's an adventure. Experience it!

Order no. 0051R tape $9.95 Requires Level II 16K.

✓2

Instant Software™

PETERBOROUGH, N.H. 03458

To Order: See your local Instant Software dealer. If these programs are unavailable, order directly.

Call Toll-Free
1-800-258-5473

✓See List of Advertisers on page 306

B (see Fig. 2). Thus, to determine direction you must use at least two photocells and poll the associated port until you detect one of the shadowed cells. If the object continues to move, it will shadow the other cell. Note: When the object is smaller than the distance between the cells, only one cell will be covered at a time; otherwise, both cells will be shadowed for some period. Also note that once one of the cells becomes uncovered, you should wait until the other cell is uncovered before trying to determine the direction of any other object.

You may wonder what would happen if one object passes through one photocell, and before that object passes through the second cell, another object passes through the first cell. This problem can be minimized by placing the photocells very close together.

Sensing Speed

Determining the approximate speed of an object moving parallel to the photocells is also possible. Once one of the sensors is shadowed, start a counter and continue to count until the other sensor is shadowed. The value of the counter will be proportional to the time it takes the object to pass between the two cells. The exact time can be computed by multiplying the counter by the time it takes to execute the counting loop. The time to execute one pass of the loop is simply the sum of the execution times of the instructions within the loop.

Since the time durations may be small, greater accuracy can be obtained by using assembly language (see Program Listing 2). The time required to execute a loop in assembly language can be obtained from the Editor/Assembler manual. For example, the instruction LD H,E takes 1.0 microsecond (a millionth of a second) to execute on a Z-80 operating at a clock speed of four megahertz. Since the TRS-80's Z-80 operates at 1.78 megahertz, this instruction would execute in approximately 2.24 microseconds. Thus, knowing the distance between the cells and the time it takes to traverse this distance will produce the approximate speed of the object. (We use the word approximate because photocells are analog devices with different response functions.) Nevertheless, the estimated speed should be sufficient for most non-critical applications.

Other Applications

These capabilities give rise to other possible applications, such as security and energy management. It seems reasonable that efficient energy management would require the heating and cooling of only those areas that are being utilized. Since it is possible to keep track of the number of persons in a region, it is possible to heat and light those areas and reduce heat and turn off the lights elsewhere. This was a very satisfying experience for the bank, which received some good information; the students, who were able to work on a real-world problem; and the authors, who were able to watch—with some pride—the TRS-80 flawlessly perform three weeks of extremely tedious work.

In these days of large scale integrations (LSI) and Very Large Scale Integration (VLSI), simple devices are sometimes overlooked. Photocells are inexpensive and simple devices; yet when combined with a microcomputer and some software, they can be of enormous use.■

Photo 1. EYE-80 In Use At Bank

like the golf pros

THE STOCKING SOURCE

comes through with lower prices

- *Your Closest Link to the Manufacturer*
- *All Items Listed are Stocked*
- *We Support & Service Products Purchased From Us*

NEW EPSON Model MX-80FT List $745.

Friction Feed has been added to the popular MX-80. All other features the same, including an adjustable tractor.

Model MX-80 .... List $645.

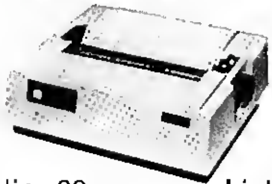
Anadex Model DP-9001 ... List $1550.
80 Column

Model DP-9501 .... List $1650.
132 Column

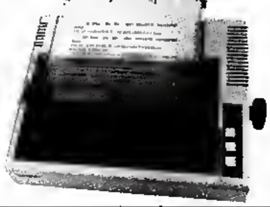
Unbelievable Dot Matrix Printer with All Functions Program-Accessible. Has 4-Character Font, Hi-Res Graphics, Switch Selectable Parallel, Serial & Current Loop Interface, Operator Selectable Protocol.

PRINTERS

Okidata —
Microline 83 .... List $1260.
Up to 15" Paper Width, 136 Columns Standard, 132 Compressed, Vertical Tab, Top of Form, Parallel & RS-232C Interfaces & Tractor Feed Included, Friction & Pin Feed.



Microline 82 .... List $799.
Microline 80 .... List $599.
Tractor Feed for
Microline 80 & 82 ...... List $50.
Hi-Speed Interface
for Microline 82 & 83 .. List $210.
256 Character Buffer, Expandable to 2K
RS-232C Interface
for Microline 80 ...... List $150.
256 Character Buffer
Epson — Model MX-70
.... List $495.
Bit Plot Graphic Printer, Adjustable Tractor 4 — 10 Inches, 80 CPS.



NEW — INTERFACES

Hi-Res Bit Plot Graphics for Epson Printers .... List $90/Set.
Plug In Prom Set

Vardon 1K Serial Interface Buffer for Epson MX-80 ... List $175.
(Complete with Cable)

Tymac — Apple Interface Using 8th Data Bit .... List $139.
(Complete with Cable)

INTERFACES

Apple Plug-In Interface
& Cable .... List $110.
Serial Interface & Cable List $90.
IEEE 488 Interface
& Cable .... List $80.
TRS-80* Expansion Interface
Cable .... List $35.
*A trade name of Radio Shack — Tandy Corp.
Model 232 Universal Serial Interface Cable .... List $25.

master charge VISA

No Penalty for Credit Card Orders

NEW — PRINTER LINES

Centronics Printers In stock.
Model 730-3 (Serial) - List $845.
Model 737-1 (Parallel) - List $995.
Model 737-3 (Serial) - List $1045.
Model 704-9 (Serial) - List $1795.
Model 704-11 - List $1870.
Also Ribbons & Accessories
All other Centronics Printers are Available.
We now stock NEC Spinwriters, Models 5510, 5515, 5520, 5525 & 5530. Also Thimbles & Ribbons. All other models are available, including Band Printers.

TEST EQUIPMENT

We stock Beckman, Fluke, Global Specialties, Keithley & NLS DMM's and probles. Also Hayes Technical Breakout Boxes.

COUPLERS & MODEMS

— Tek-Com

FURNITURE & PRINTER STANDS

Systems Furniture and Printer manufacturers

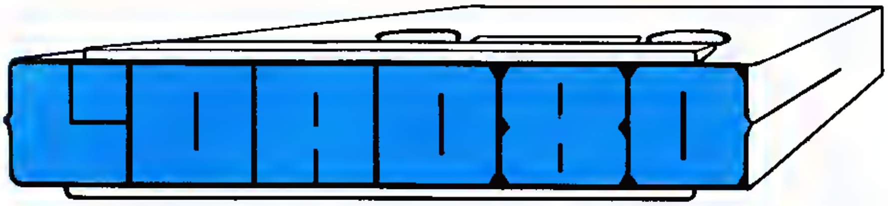
ASK FOR OUR INSTANT DISCOUNT
From Roy Hawthorne
Talk To Bill Tokar On Applications

CALL TOLL FREE
U.S.A.
1-800-521-2764
MICHIGAN
1-800-482-8393

Reminder:
We are open
8:30 to 6:00 PM EST
Monday through Saturday

WRITE TO: ✓438
"The Stocking Source"
24069 Research Drive
Farmington Hills, MI 48024
313-474-6708

80 MICROCOMPUTING ANNOUNCES...



. . . and saves you hours of typing and aggravation.

LOAD-80 is a monthly dump of the major program listings in 80 Microcomputing on cassette. Publisher Wayne Green tells you more. . .

"Frankly, after hundreds of hours of frustration, I seldom even try to keyboard a published program. Even if the magazine manages to get the program typeset correctly (which seems rare), I inevitably screw it up when I keyboard it. Who needs the aggravation?

"This is why I've started a new series of cassettes called **Load 80**. Each cassette will have program dumps of the listings in an issue of **80 Microcomputing**. These listings are direct from the authors and tested by the 80 staff. All but the very short program listings will be on these **Load 80** cassettes. Thus you will be able to save hours of inputting programs and even more of debugging your keyboarding errors.

"Though the authors of these programs will share the royalties from the sale of the cassettes, this will not preclude the better programs from being issued separately by Instant Software (with royalties) with full documentation and associated hoopla. The documentation for the **Load 80** programs will be entirely in **80 Microcomputing.**

"I originally was holding out for "Trash Dump" as a name for the cassettes, but cooler heads prevailed. If there turns out to be enough interest in **Load 80**, we'll set up a monthly subscription arrangement."

Wayne Green, Publisher

Send in the attached card and you will receive the cassette for the major programs in this issue (June 1981). If the card is gone, photocopy the coupon.

Please note there is no warranty expressed or implied that this program is going to do anything other than save you typing

☐ Yes. . . send my LOAD 80 Cassette for only $9.95

☐ Check Enclosed Bill my ☐ AE ☐ MC ☐ VISA

NAME ____________________

ADDRESS ____________________

CITY __________ STATE ________ ZIP ________

CARD # ______________ EXPIRE DATE ________

SIGNATURE ______________ INTERBANK # ________

Please allow 4-6 weeks for delivery
80 Microcomputing Pine Street Peterborough NH 03458
Attn: Debra L. Boudrieau

316B8

Wherein our author discovers the intimate relationship between graphic elements and their ASCII character codes.

Unlocking the Graphic Code

Jerome I. Weintraub
690 Mountain View Rd.
El Cajon CA 92021

One evening as I was hard at work creating an educational program for the students at my school, I had occasion to use one of the graphic elements listed within ASCII character codes 129-191, described on page C/2 of the Level II BASIC Reference Manual. It seems that every time I needed to use one of the characters, I had to set up the program described in the manual, with minor modifications, in order to select the one I wanted. The program:

```
10 FOR X = 129 TO 191
20 PRINT CHR$(23) : PRINT
30 PRINT X; : PRINT CHR$(X)
40 FOR Y = 1 TO 500 : NEXT Y
50 NEXT X
```

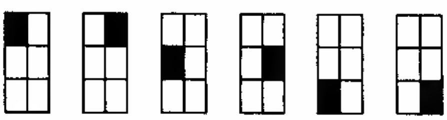
I seldom take the time to establish a routine that will probably save a good deal of time in the long run. I usually put it off until some indefinite future date, then end up reinventing the wheel time and again as punishment for having neglected to do it once and for all.

This time I decided to take the time to write or draw all the codes on a sheet of programming paper. Never again would I have to hunt and peck through the graphic elements displayed one by one on my screen.

There are 63 different graphic designs, numbered 129 through 191. As I copied them from the screen onto the programming sheet, a certain symmetry began to appear. Gradually I was able to predict what the next design would look like, and finally I was able to draw them without having to check the screen before or after drawing them (although I must confess I peeked, in order to be satisfied they were correct).

The main reason I discovered this is that I renumbered the designs, from 1 to 63. I then noticed that 17 was exactly the same as 1 and 16 combined; 33 was the same as 1 and 32 combined (Fig. 1).

In other words, $17 = 1 + 16$; $33 = 1 + 32$; and $59 = 10 + 49$. As I studied this development, I searched for those pips that appeared alone within the six-sectored grid. Not surprisingly, I found the following:



1 2 4 8 16 32

Not only did I find perfect symmetry, but it followed the basis of all computer technology: the binary code!

The final two developments followed in rapid succession. First, I drew a composite scheme of all six locations, noting the value appropriate to each sector:

1	2
4	8
16	32

Finally, I realized I could create any arrangement I desired by selecting the sectors I wanted to use and calculating the appropriate ASCII code. For example, if I wanted a vertical bar on the right side of the grid, I would add $2 + 8 + 32$, equal to 42, and add 42 to 128 to find the ASCII character code of 170 for the desired arrangement.

Where did I get the 128? The first ASCII graphic code is 129. It produces arrangement 1. $129 - 1$ equals 128! All you need to do is:

- Design the arrangement you want on Fig. 3.
- Add up the values of the sectors you selected.
- Add this sum to 128.
- Enter PRINT CHR$(YOUR SUM)

I learned two lessons from this experience. First, of course, was the symmetry and mathematical basis for the graphic code. But more important, perhaps, is the valuable but time-consuming job of documenting routines I have developed. Recording and cataloging your reusable routines will repay generously the time it takes you to do the job right. ■

1 + 16 = 17 1 + 32 = 33 10 + 49 = 59

Fig. 1.

H & E COMPUTRONICS INC.

•••EVERYTHING FOR YOUR TRS-80™•••

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation

BUSINESS PAC 100

NEW

★ All orders processed within 24-Hours
★ 30-Day money back guarantee on all Software

100 Ready-To-Run Business Programs

(ON CASSETTE OR DISKETTE).....Includes 110 Page Users Manual.....5 Cassettes (Or Diskettes)
Inventory Control.....Payroll.....Bookkeeping System.....Stock Calculations.....
Checkbook Maintenance.....Accounts Receivable.....Accounts Payable.....

BUSINESS 100 PROGRAM LIST

NAME	DESCRIPTION
1 RULE78	Interest Apportionment by Rule of the 78's
2 ANNUI	Annuity computation program
3 DATE	Time between dates
4 DAYYEAR	Day of year a particular date falls on
5 LEASEINT	Interest rate on lease
6 BREAKEVN	Breakeven analysis
7 DEPRSL	Straightline depreciation
8 DEPRSY	Sum of the digits depreciation
9 DEPRDB	Declining balance depreciation
10 DEPRDDB	Double declining balance depreciation
11 TAXDEP	Cash flow vs. depreciation tables
12 CHECK2	Prints NEBS checks along with daily register
13 CHECKBK1	Checkbook maintenance program
14 MORTGAGE/A	Mortgage amortization table
15 MULTMON	Computes time needed for money to double, triple, etc.
16 SALVAGE	Determines salvage value of an investment
17 RRVARIN	Rate of return on investment with variable inflows
18 RRCONST	Rate of return on investment with constant inflows
19 EFFECT	Effective interest rate of a loan
20 FVAL	Future value of an investment (compound interest)
21 PVAL	Present value of a future amount
22 LOANPAY	Amount of payment on a loan
23 REGWITH	Equal withdrawals from investment to leave 0 over
24 SIMPDISK	Simple discount analysis
25 DATEVAL	Equivalent & nonequivalent dated values for oblig.
26 ANNUDEF	Present value of deferred annuities
27 MARKUP	% Markup analysis for items
28 SINKFUND	Sinking fund amortization program
29 BONDVAL	Value of a bond
30 DEPLETE	Depletion analysis
31 BLACKSH	Black Scholes options analysis
32 STOCVAL1	Expected return on stock via discounts dividends
33 WARVAL	Value of a warrant
34 BONDVAL2	Value of a bond
35 EPSEST	Estimate of future earnings per share for company
36 BETAALPH	Computes alpha and beta variables for stock
37 SHARPE1	Portfolio selection model-i.e. what stocks to hold
38 OPTWRITE	Option writing computations
39 RTVAL	Value of a right
40 EXPVAL	Expected value analysis
41 BAYES	Bayesian decisions
42 VALPRINF	Value of perfect information
43 VALADINF	Value of additional information
44 UTILITY	Derives utility function
45 SIMPLEX	Linear programming solution by simplex method
46 TRANS	Transportation method for linear programming
47 EOQ	Economic order quantity inventory model
48 QUEUE1	Single server queueing (waiting line) model
49 CVP	Cost-volume-profit analysis
50 CONDPROF	Conditional profit tables
51 OPTLOSS	Opportunity loss tables
52 FQUOQ	Fixed quantity economic order quantity model
53 FQEOWSH	As above but with shortages permitted
54 FQEOQPB	As above but with quantity price breaks
55 QUEUECB	Cost-benefit waiting line analysis
56 NCFANAL	Net cash-flow analysis for simple investment
57 PROFIND	Profitability index of a project
58 CAP1	Cap. Asset Pr. Model analysis of project
59 WACC	Weighted average cost of capital
60 COMPBAL	True rate on loan with compensating bal. required
61 DISCBAL	True rate on discounted loan
62 MERGANAL	Merger analysis computations
63 FINRAT	Financial ratios for a firm
64 NPV	Net present value of project
65 PRINDLAS	Laspeyres price index
66 PRINDPA	Paasche price index
67 SEASIND	Constructs seasonal quantity indices for company
68 TIMETR	Time series analysis linear trend
69 TIMEMOV	Time series analysis moving average trend
70 FUPRINF	Future price estimation with inflation
71 MAILPAC	Mailing list system
72 LETWRT	Letter writing system-links with MAILPAC
73 SORT3	Sorts list of names
74 LABEL1	Shipping label maker
75 LABEL2	Name label maker
76 BUSBUD	DOME business bookkeeping system
77 TIMECLCK	Computes weeks total hours from timeclock info.
78 ACCTPAY	In memory accounts payable system-storage permitted
79 INVOICE	Generate invoice on screen and print on printer
80 INVENT2	In memory inventory control system
81 TELDIR	Computerized telephone directory
82 TIMUSAN	Time use analysis
83 ASSIGN	Use of assignment algorithm for optimal job assign.
84 ACCTREC	In memory accounts receivable system-storage ok
85 TERMSPAY	Compares 3 methods of repayment of loans
86 PAYNET	Computes gross pay required for given net
87 SELLPR	Computes selling price for given after tax amount
88 ARBCOMP	Arbitrage computations
89 DEPRSF	Sinking fund depreciation
90 UPSZONE	Finds UPS zones from zip code
91 ENVELOPE	Types envelope including return address
92 AUTOEXP	Automobile expense analysis
93 INSFILE	Insurance policy file
94 PAYROLL2	In memory payroll system
95 DILANAL	Dilution analysis
96 LOANAFFD	Loan amount a borrower can afford
97 RENTPRCH	Purchase price for rental property
98 SALELEAS	Sale-leaseback analysis
99 RRCONVBD	Investor's rate of return on convertable bond
100 PORTVAL9	Stock market portfolio storage-valuation program

☐ **CASSETTE VERSION** **$99.95**
☐ **DISKETTE VERSION** **$99.95**
☐ **TRS-80* MODEL II VERSION** **$149.95**

ADD $3.00 FOR SHIPPING IN UPS AREAS
ADD $4.00 FOR C.O.D. OR NON-UPS AREAS
ADD $5.00 OUTSIDE U.S.A., CANADA & MEXICO

NEW TOLL-FREE ORDER LINE
(OUTSIDE OF N.Y. STATE)
(800) 431-2818

COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

VISA

master charge

24 HOUR ORDER LINE
(914) 425-1535

H & E COMPUTRONICS INC.

•••EVERYTHING FOR YOUR TRS-80™•••

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation

PAYROLL

Payroll invoices many complex calculations and the production of reports and documents, many of which are required by government agencies. It is an ideal candidate for the computer. With this Payroll system in-house, you can promptly and accurately pay your employees and generate accruate documents/reports to management, employees, and appropriate government agencies concerning earnings, taxes, and other deductions. The package has been converted to the TRS-80™ and is now a well documented, op-line, interactive, micro-computer system with the capabilities of (or exceeding) many larger systems.

CAPABILITIES:

★ performs all necessary payroll tasks including:
- file maintenance, pay data entry and verification
- computation of pay and deduction amounts
- printing of reports and checks

★ can handle salaried and hourly employees

★ employees can receive:
- hourly or salary wage
- vacation pay
- holiday pay
- piecework pay
- overtime pay

★ employees can be paid using any combination of pay types (except, hourly cannot receive salary and salary cannot receive hourly)

★ special non-taxable or taxable lump sums can be paid regularly or one time (bonus, reimbursements, etc)

★ health and welfare deductions can be automatically calculated for each employee

★ earnings-to-date are accumulated and added to permanent records; taxes are computed and deducted: US income tax, Social Security tax, state income tax, other deductions (regular or one time)

★ paychecks are printed; computer checks with your company letterhead can be purchased from SBSG

★ calculations are accumulated for; employee pay history, 941A report, W-2 report, insurance report, absentee report

★ fully linked to General Ledger. Each employee's payroll information can be distributed to as many as (12) twelve different GL accounts; system automatically posts to cash account

INVENTORY CONTROL/INVOICING

★ **ISAM** (Indexed Sequential Access Method) eliminates the necessity for time consuming sort.

★ Pre-Allocated Files for IMMEDIATE update and inquiry capabilities.

★ Fast Disk storage and retrieval.

★ Inventory Master Record includes...class...SKU...Division...Retail... Cost...Beginning Balance...Period Sale Units...Period Receipts...On Order...On Hand...Minimum Reorder Point...Recommended Reorder Amount...Vendor Number...Period Sale Dollars...YTD Sale Units...YTD Sale Dollars.

★ Calculated and Displayed Formulas include...Gross Margin ($)... Gross Margin (%)...Gross Margin ROI (%)...Average Inventory Retail ($)...Average Inventory Cost ($)...Turn-Over (%).

★ Reports Generated include...Master File Listing...Class Description Listing...Transaction Audit Trail...Minimum Reorder Point by Vendor...Retail Price List...Retail & Cost Price List...Period Sales Report ...Year to Date Sales Report...Stock Status (Screen or printer output) ...Commission Report (for salesmen and buyers).

★ Transaction Types include...Sales, Vendor Receipts...Vendor Orders...Customer Returns...Vendor Returns...Transfer Stock.

GENERAL LEDGER

The General Ledger accounting system consolidates financial data from other accounting subsystems (A/R, A/P, Payroll, direct posting) in an accurate and timely manner. Major reports include the Income Statement and Balance Sheet and a "special" report designed by management. The beauty of this General Ledger system is that it is completely user formatted. You "customize" the account numbers, descriptions, and report formats to suit particular business requirements. These programs were developed 5 years ago for the Wang micro-computer and have been tested in many environments since then. The package has been converted to the TRS-80™ and is now a well documented, on-line, interactive micro-computer system with the capabilities of (or exceeding) many larger systems.

CAPABILITIES:

★ more than 200 chart of accounts can be handled

★ account number structure is user defined and controlled

★ more than 1,750 transactions may be entered via:
- direct posting: done by hand; validated against the account file before acceptance
- external posting; generated by A/R, A/P, Payroll or any other user source

★ data is maintained and reported by:
- month
- quarter
- year
- previous three quarters

★ reports (samples on back) include:
- trial balances
- income statement
- balance sheet
- special accounts reports and more.....

★ user formats reports with the following designated as you wish:
- titles
- headings
- account numbers
- descriptions
- subtotals
- totals
- skip lines
- skip pages

★ up to eight levels of totals - fully user designated

★ menu driven; easy to use; full screen prompting and cursor control

MICRO COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

NEW TOLL-FREE ORDER LINE
(OUTSIDE OF N.Y. STATE)
(800) 431-2818

24 HOUR ORDER LINE
(914) 425-1535

VISA — master charge

PRICING	MOD-1 VERSION	MOD-II VERSION	MOD-III VERSION
ACCOUNTS RECEIVABLE	$125	$225	$199.95
ACCOUNTS PAYABLE	$125	$225	$199.95
GENERAL LEDGER	$125	$225	$199.95
PAYROLL	$125	$225	$199.95
INVENTORY	$175	$275	$199.95
INVOICING	$150	$250	$199.95
COORDINATED INVENTORY/INVOICING ACCOUNTS RECEIVABLE	$449	$749	$599.95
COORDINATED AR-AP-GL	$375	$675	$599.95
COORDINATED AR-AP-GL with PAYROLL	$495	$899	$799.95
EXTENDED COORDINATED AR-AP-GL INVOICING/INVENTORY with PAYROLL	$799	$1299	$1199.95

See List of Advertisers on page 306

Every gamesman needs them.

Joysticks for the Model I

Frank DiNunzio
309 Westminster Ave.
Bristol, PA 19007

About six months ago I built a two joystick interface unit for my TRS-80.

Construction was simple and practically foolproof.

The entire system consists of two joysticks, each mounted in a plastic box. A push button is included as a Fire button. Another box has the interface board and connecting cables. It plugs into the expansion port in the rear of the computer. No modifications to the TRS-80 itself are necessary.

Power to operate the interface is supplied by a self-contained battery that is turned on when the edge connector is plugged into the expansion port.

The circuit uses two tri-state hex inverters (74LS368), that are turned off and on alternately with a 74LS73 dual JK flip-flop. The flip-flop toggles each time an INP (1) command is used. If any of the inverters' inputs are at ground potential when the 74LS368 is on, the corresponding data is sent to the CPU.

The joystick pots are used as on-off switches, to ground the inputs of the inverters. The push button switch (Fire), shorts all four inputs to ground simultaneously, through the four diodes.

A program line such as: 20 A=INP(1): B=INP(1) operates the interface by returning values for A and B to the CPU. It is important that two INP statements are used on the same line so the flip-flop operates twice when that part of the program executes. Otherwise the joysticks will be out of sync.

These parts were used because they were readily available.

● Joysticks: substitutes must

Quantity	Part	Manufacturer
2	Tri State Hex Inverter	Radio Shack #276-1836
1	Dual JK Flip-Flop	Radio Shack #276-1918
3	Experimenters Box	Radio Shack #270-233
2	Terminal Strip	Radio Shack #274-688
2	Pushbutton Switch	Radio Shack #275-011
1	5 Conductor Cable	Radio Shack #42-2151
8	Diodes	Radio Shack #276-1101
8	4.7 kilohm Resistors	Radio Shack #271-1330
1	Circuit Board	Radio Shack #26-170
1	Battery Holder	Radio Shack #270-383
2	200 kilohm Joystick TM21K167	Herbech and Rademan, Philadellphia, PA
1	40 Conductor Cable 2037	Hobbyworld Electronics Northridge, CA
2	16-Pin IC Sockets	Radio Shack #276-1998
1	14-Pin IC Socket	Radio Shack #276-1999
4	AA Batteries	

Parts List

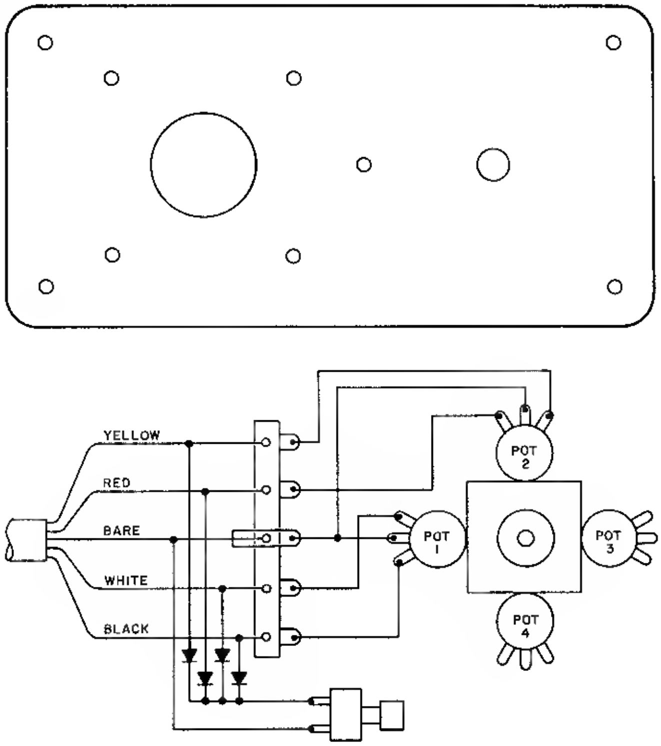


Fig. 1

be able to go down to 0 Ohms resistance.

- Experimenter's Box.
- Five-Conductor Cable: the plugs must be cut off, and the cable cut into two equal lengths. A second option is to leave the plugs on the cable and add two DIN sockets to the interface box. This will allow the joysticks to be unplugged and permit other input devices to control the interface unit.
- Circuit Board: this must be cut off at about the 30th row to fit into the experimenter's box.
- 40-Conductor Ribbon Cable: check the wiring to see where they terminate on the edge connector. Most are wired 2, 1, 4, 3, 6, 5 etc. with respect to the TRS-80 edge plug. Also, be sure to mark the top of the connector, because it can be put in upside-down.
- Broken lines in Fig. 2: indicate the wire jumpers that go on the underside of the PC board where the foil is.
- IC Sockets: some of the mounting pins have to be cut off the sockets to simplify wiring of the circuit board. The pins not needed on the 16 pin sockets are: 11, 12, 13, 14, and 15. The pins to remove on the 14 pin socket are: 1, 2, 3, 6, 7, 10, 12, 13, and 14. They can be bent up instead of cut.

Assembling the Joysticks

Using Fig. 1 as a template, cut the mounting holes in the cover of one of the boxes. Mount the joystick, terminal strip, and push button switch to the underside of the cover. Connect a wire to the center tap of pot #2 and run it to the center tap of pot #1 then to terminal #3 on the terminal strip. (See Fig. 2). Solder a wire to the bottom tap of pot #1 to terminal #5 on strip. Fasten a wire to the top tap of pot #1 to terminal #4. Connect a wire to the top tap on pot #2 then to terminal #1. Run a wire from the bottom tap of pot #2 to terminal #2. Fasten a diode to each of the terminals (1, 2, 4, and 5) on the strip. The other ends, with the band around them, are soldered to one side of the push button switch. The other side of the switch is wired to terminal #3 on the terminal strip.

Connect the five-conductor cable as follows:

- Yellow - terminal #1
- Red - #2
- Bare wire - #3
- White - #4
- Black - #5

Cut a groove in the edge of the bottom part of the box at the end for the cable to go through when the cover is re-assembled.

This completes construction of one controller. Repeat the instructions for the second one. They are identical.

Assembling the Interface Unit

Take both 16 pin IC sockets and cut off pins 11, 12, 13, 14, and 15. From the 14 pin socket remove pins 1, 2, 3, 6, 7, 10, 12, 13 and 14. Insert the IC sockets as shown in Fig. 3. This is the top side of the Radio Shack board (#276-170). Turn the board over and solder jump wires on the bottom as follows: E2 to F2, E4 to F4, E6 to F6, E8 to F9, E11 to F11, E13 to F13, E15 to F15, E22 to F17, E23 to F22 and A23 to X23.

Turn the board over again and solder jumpers from A10 to A24, D8 to D17, D1 to D25. Run wires from G8 to G17, G10 to G23. Connect H1 to H10, H7 to H16. Attach jumpers from I3 to I12, I5 to I14. Install the 4.7 k Ohm resistors from X2 to B2, X4 to B4, X5 to B6, X7 to B7, X11 to B11, X13 to B13, X15 to B15, and X16 to B16. Connect the +5 volt battery lead to J22 and the −5 volt lead to Y22. Attach the 40 connector jumper cable to the board as follows:

- J1 to 19 on edge connector
- J3 to 30
- J5 to 22
- J7 to 32
- J8 to 37
- J9 to 26
- Y10 to 29

Connect the first joystick to the board as follows:

- Yellow wire to C2
- White wire to C4
- Red wire to C6
- Black wire to C7
- Bare copper wire to C22

The second joystick is wired as follows:

- Yellow to C11
- White to C13
- Red to C15
- Black to C16
- Bare wire to C22

File two grooves for the five-conductor cable in the edge of the bottom section of the experimenters box. Also, file a slot for the ribbon that goes to the 40 pin edge connector. Fasten the battery holder to the bottom side of the board with double-sided tape, insert batteries, ICs, then attach lid.

This completes the assembly

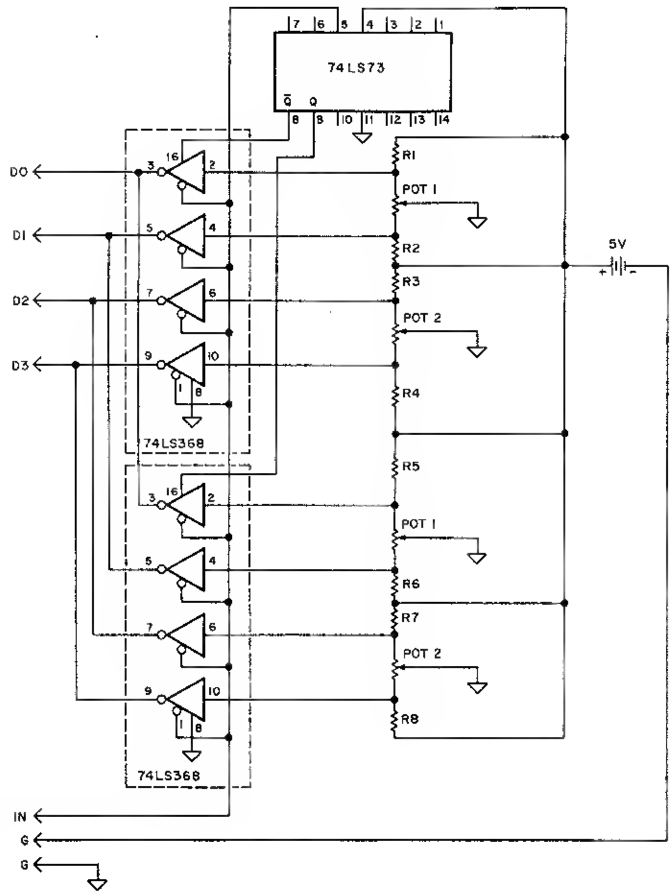


Fig. 2

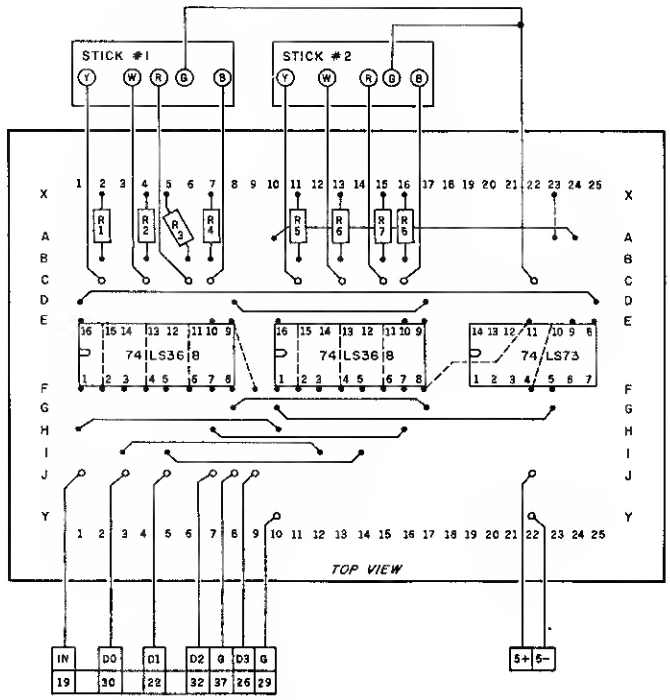


Fig. 3

of the entire system.

Testing the Joystick Controllers

Plug the 40 pin edge connector into the port in the rear of the TRS-80. The battery is automatically connected to the interface through the grounds of the TRS-80 bus when the connector is attached.

The system is now ready for testing. Type in the program listed below:

```
10 CLS
20 A = INP (1): B = INP (1)
30 PRINT"JOYSTICK A = ";A,
40 PRINT"JOYSTICK B = ";B
50 GOTO 20
```

Set the joystick handles to the center position and run the program. The values for A and B should be 240. Move the handles to different positions and the numbers will change values to:

UP = 244; DOWN = 241; LEFT = 242; RIGHT = 248; PUSHBUTTON = 255; UP and RIGHT = 252; DOWN and RIGHT = 249; DOWN and LEFT = 243; UP and LEFT = 246;

Here is a demonstration program that will draw lines on the CRT.

```
10 CLS
20 A = INP(1): B = INP(1)
30 PRINT@0,A;
40 IF A = 241 THEN Y = Y + 1: GOTO 90
50 IF A = 244 THEN Y = Y − 1: GOTO 90
60 IF A = 242 THEN X = X − 1: GOTO 90
70 IF A = 248 THEN X = X + 1: GOTO 90
80 IF A = 255 THEN 10
90 IF X>127 THEN X = 127
100 IF X<1 THEN X = 1
110 IF Y>47 THEN Y = 47
120 IF Y<1 THEN Y = 1
130 SET (X,Y)
200 GOTO 20
```

This program operates with joystick #1 and is similar to Etch-a-Sketch. The push button will erase the screen. The numbers at the corner of the screen are the values returned by the joysticks to the computer. They are for reference only.

Both joysticks can be used in a program by assigning the value returned by A to one player and the B value to a second player. ■

Arrays give you a perspective on mazes in this 3-D game.

SUPERMAZE

Howard F. Batie
12002 Cheviot Drive
Herndon, VA 22070

Supermaze puts you inside a maze looking down a corridor. The corridor lies ahead of you in complete perspective, while halls lead off to the right and left. (See Fig. 1.) It's up to you to guess which way to go.

In the maze you can see a maximum distance of four units ahead. However, if there's a wall three squares ahead, you can't see beyond it.

Three Options

Each move offers three options: forward one, left or right.

After each move you get a new picture of what lies ahead. A counter keeps track of the number of forward moves, but is not incremented if you turn. The minimum number of forward moves to successfully exit is printed along with your score when you leave the maze.

If you get turned around and leave the maze at the entrance, you lose. And if you're unfortunate enough to walk into an electric wall, you fry.

Listing 1 has six mazes of increasing difficulty and is written for a Level II TRS-80 with 16K.

Table 1 shows the logic flow

Line	Operation
80	Clear Screen DIMension Arrays
200	Enter maze at A(100)
290	Start: GOSUB 830
830	Set Y for direction you are facing
870	Fetch Array A contents for your current location
880	Rotate Array A contents for your current direction: GOTO 300
300	Is there a wall to the right?
320	Yes: Draw wall: GOTO 360
350	No: Draw hall
360	Is there a wall to the left?
380	Yes: Draw wall: GOTO 420
410	No: Draw hall
420	Is there a wall ahead?
460	Yes: Draw wall: GOTO 500
470	No: E = E + 1: Is E>4?
470	Yes: GOTO 500
480	No: GOSUB 830 (Set Y for next location forward)
500	Move Forward or Turn (Type F, R or L)
550	Move Forward (Type F)
550	Did I exit at the entrance?
550	Yes: GOTO 700
550	No: Proceed
560	Did I leave at the exit?
560	Yes: GOTO 760
560	No: Proceed
580	Did I run into a wall?
580	Yes: GOTO 710
600	No: GOSUB 830
620	Turn Right (Type R)
620	D = D + 1: IF D>4 THEN D = 1: GOSUB 830
660	Turn Left (Type L)
660	D = D − 1: IF D<1 THEN D = 4: GOSUB 830
700	Clear Screen: Lose: Print Message: END
710	Clear Screen: Lose: Print Message: END
760	Clear Screen: Win: Print Message: END

Table 1. Supermaze Logic Flow Chart

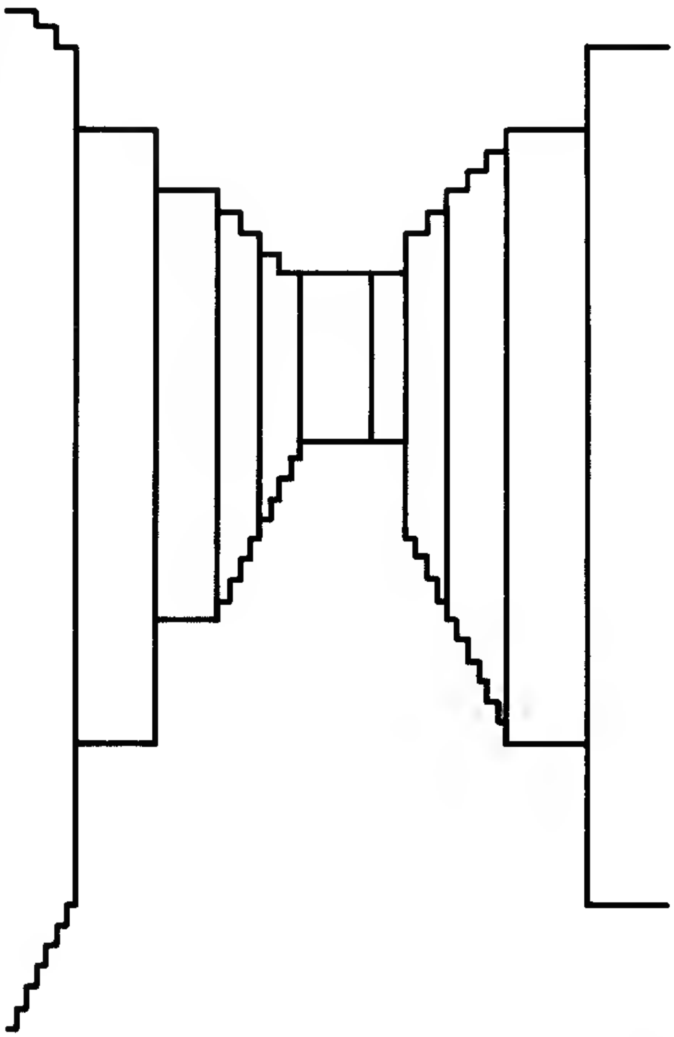


Fig. 1.

in a form that is easier (for me, anyway) to understand than a flowchart. The logic flows downward unless there is a branching instruction.

The program's first array is called A and has the dimensions of 105 × 1. It uses the zero element. The first 100 elements (0–99) contain either a zero value or a five-digit decimal number which defines the shape of its maze location. Visualize the first 100 elements of the A array as a 10 × 10 matrix (which is the maximum size of the maze) shown in Fig. 2.

Constructing the Maze

The program statements 930-980 in Listing 1 construct the 8 × 7 maze of Fig. 3 with the entrance at location 60 and the exit at location 57. Other mazes shown in Figs. 4 through 8 are constructed by statements 990–1270. Note that all mazes must be entered from the left and exited on the right side because the initial direction (D) is equal to 2 in line 200.

The final five elements of the A array (100-104) specify the starting and ending locations, minimum number of moves to the exit, and the size of the maze. These can differ for each maze.

If any numbered matrix location in the grid is outside the maze, the contents of the corresponding element will be set to zero; otherwise the five-digit decimal defines the shape of the maze location. To prevent blanking of leading zeros in the last four digits, the first of the five digits is always one.

In each of the last four digits, a one represents a wall and a zero represents a hall (no wall). The second, third, fourth and fifth digits correspond to the north, east, south and west sides. For example, the shape of block 60 in Fig. 3 is designated by 10000, and block 65 is designated by 10101.

Change the Shape

To change the shape of the maze, simply code the data statements to correspond with the particular maze you con-

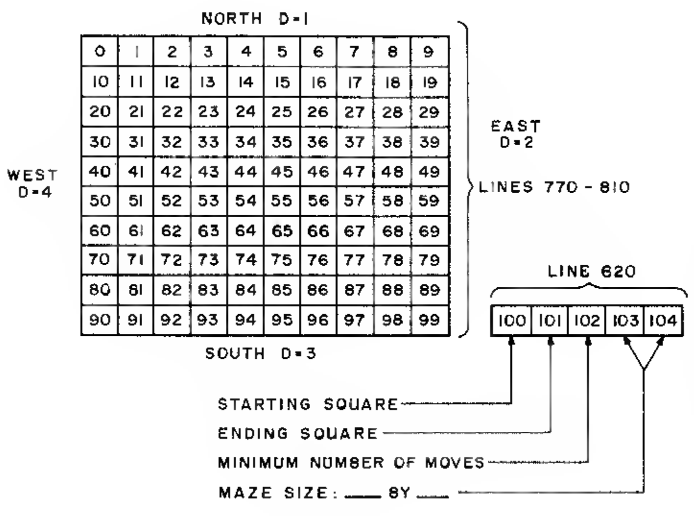


Fig. 2.

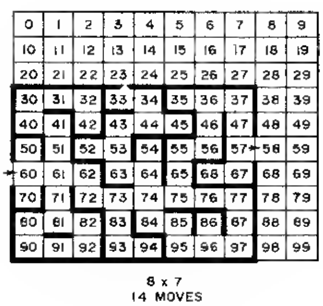


Fig. 3.

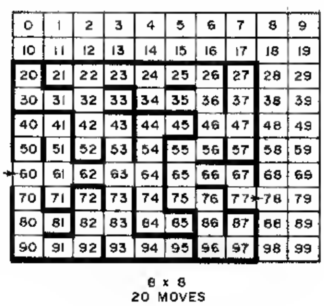


Fig. 4.

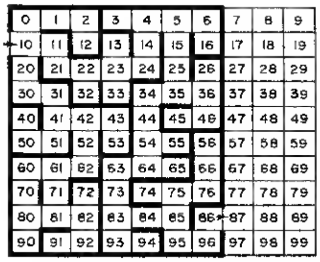
Program Listing. Supermaze

```
30 CLS:PRINT@33Ø,"*  *  *   S  U  P  E  R  M  A  Z  E *"
40 PRINT@7Ø4,"COPYRIGHT 198Ø"
5Ø PRINT"BY HOWARD F. BATIE"
6Ø PRINT"HERNDON, VA"
7Ø FORI=1TO1ØØØ:NEXTI
75 REM -- SETUP
8Ø CLEAR:DIMA(1Ø4),D(23)
9Ø FORI=ØTO23:READD(I):NEXTI
11Ø CLS:PRINT"   WHICH MAZE DO YOU WANT"
12Ø LL=Ø:FORI=ØTO5:PRINT@128*I+197,"MAZE NR";D(4*I+LL);
13Ø PRINT"IS";D(4*I+LL+1);"BY";D(4*I+LL+2);
14Ø PRINTTAB(26),"MINIMUM NUMBER OF MOVES IS";D(4*I+LL+3)
142 LL=Ø:NEXTI
15Ø PRINT@25,"";:INPUTMN
16Ø IFMN>6ORMN<1PRINT@25,"      ";:GOTO15Ø
17Ø CLS:IFMN=1THEN19Ø
18Ø FORI=ØTO(1Ø5*(MN-1))-1:READAA:NEXTI
19Ø FORI=ØTO1Ø4:READA(I):NEXTI
2ØØ E=Ø:X=A(1ØØ):D=2
2Ø5 REM -- INSTRUCTIONS
21Ø CLS:PRINT@128,"YOU ARE IN A";A(1Ø3);"BY";A(1Ø4);
212 PRINT"MAZE WITH ELECTRIFIED WALLS.  FIND YOUR"
22Ø PRINT"WAY OUT IN THE LEAST NUMBER OF MOVES.":PRINT
23Ø PRINT"THE MINIMUM NUMBER OF MOVES IS";A(1Ø2);
232 PRINT"FOR THIS MAZE.":PRINT
24Ø PRINT"MOVE FORWARD BY TYPING 'F', TURN RIGHT BY
    TYPING 'R'"
25Ø PRINT"OR TURN LEFT BY TYPING 'L'.":PRINT
26Ø PRINT"TURNS TO THE RIGHT OR LEFT DO NOT COUNT AS
    MOVES."
27Ø PRINT:PRINT"PRESS 'ENTER' WHEN READY TO START."
28Ø M$=INKEY$:IFM$=""THEN28Ø
285 REM -- START
29Ø CLS:GOSUB83Ø
295 REM -- IS THERE A WALL TO THE RIGHT?
3ØØ PRINT@435,"MOVES:";:PRINT@5ØØ,Q;
3Ø1 ONE+1GOTO3Ø2,3Ø7,31Ø,313,317
3Ø2 GL=15374:GR=15399
3Ø3 POKEGR,16Ø:POKEGL,144
3Ø4 GR=GR+64:GL=GL+64:IFGR>1624ØTHEN3Ø6
3Ø5 POKEGR,17Ø:POKEGL,149:GOTO3Ø4
3Ø6 POKE1627Ø,133:POKE16295,138:GOTO321
3Ø7 GL=15442:GR=15459
3Ø8 GL=GL+64:GR=GR+64:IFGR>161ØØTHEN321
3Ø9 POKEGL,149:POKEGR,17Ø:GOTO3Ø8
31Ø GL=155Ø9:GR=1552Ø
311 GL=GL+64:GR=GR+64:IFGR>1597ØTHEN321
312 POKEGL,149:POKEGR,17Ø:GOTO311
313 GL=15575:GR=15582:POKEGL,144:POKEGR,16Ø
314 GL=GL+64:GR=GR+64:IFGR>1584ØTHEN316
315 POKEGL,149:POKEGR,17Ø:GOTO314
316 POKEGL,133:POKEGR,138:GOTO321
317 GL=15641:GR=15644
318 POKEGL,148:POKEGR,168:POKE15833,129:POKE15836,13Ø
319 GL=GL+64:GR=GR+64:IFGR>15773THEN321
32Ø POKEGL,149:POKEGR,17Ø:GOTO319
321 IFVAL(MID$(B$,4,1))=ØTHEN339
322 REM --- DRAW WALL TO THE RIGHT
323 ONE+1GOTO324,326,33Ø,333,335
324 POKE154ØØ,184:POKE154Ø1,142:POKE154Ø2,131:POKE16296,
    18Ø
325 POKE1636Ø,13Ø:POKE16361,173:POKE16362,144:GOTO36Ø
326 POKE15524,131:POKE1546Ø,16Ø:POKE15461,184
327 POKE15462,142:POKE15463,171:POKE161ØØ,144
328 POKE16164,139:POKE16165,18Ø:POKE16229,13Ø
329 POKE1623Ø,173:POKE16231,186:POKE16295,139:GOTO36Ø
33Ø POKE15585,131:POKE15521,16Ø:POKE15522,184
331 POKE15523,174:POKE15969,144:POKE16Ø33,139
332 POKE16Ø34,18Ø:POKE16Ø98,13Ø:POKE16Ø99,175:GOTO36Ø
333 POKE15583,184:POKE15584,174:POKE159Ø3,18Ø
334 POKE15967,13Ø:POKE15968,175:GOTO36Ø
335 POKE15645,142:POKE15646,171
336 POKE15836,13Ø:POKE15837,173:POKE15838,186
337 POKE159Ø2,139:GOTO36Ø
338 REM --- DRAW HALL TO THE RIGHT
339 ONE+1GOTO34Ø,343,346,348,35Ø
34Ø GT=15399:GB=16295
341 GT=GT+1:GB=GB+1:IFGB=163Ø1THEN36Ø
342 POKEGT,176:POKEGB,14Ø:GOTO341
343 POKE15524,131:POKE15525,131:POKE15526,131
344 POKE15527,171:POKE161ØØ,176:POKE161Ø1,176
345 POKE161Ø2,176:POKE161Ø3,186:GOTO36Ø
346 POKE15585,131:POKE15586,131:POKE15587,171
347 POKE15969,176:POKE1597Ø,176:POKE15971,186:GOTO36Ø
348 POKE15583,176:POKE15584,186:POKE159Ø3,14Ø
349 POKE159Ø4,174:GOTO36Ø
35Ø POKE15645,14Ø:POKE15646,174:POKE15837,131
351 POKE15838,171
355 REM -- IS THERE A WALL TO THE LEFT?
36Ø IFVAL(MID$(B$,6,1))=ØTHEN377
361 REM --- DRAW WALL TO THE LEFT
362 ONE+1GOTO363,366,37Ø,373,375
363 POKE15373,18Ø:POKE15372,141:POKE15371,131
364 POKE16269,184:POKE16333,129:POKE16332,158
```

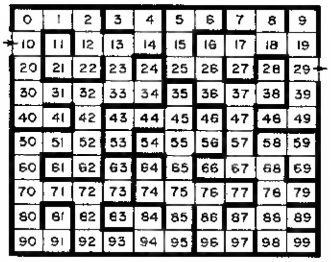
program continues

struct. Lines 990-1040 and 1050-1100 in Table 1 correspond to the mazes of Figs. 4 and 5. After you have created a number of mazes, video prompts let you choose one of six to replay and run.

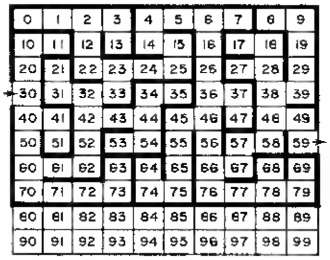
Of course, nearly every program written can be refined, and this one is no exception. Two improvements that come to mind are a built-in random maze generator and machine language graphics with more speed.■



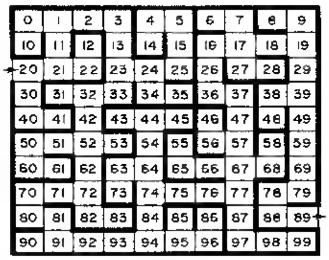
7 x 10
23 MOVES
Fig. 5.



10 x 10
36 MOVES
Fig. 7.



10 x 8
29 MOVES
Fig. 6.



10 x 10
45 MOVES
Fig. 8.


```
365 POKE16331,160:GOTO420
366 POKE15438,151:POKE15439,141:POKE15440,180
367 POKE15441,144:POKE15505,131:POKE16270,135
368 POKE16206,181:POKE16207,158:POKE16208,129
369 POKE16144,184:POKE16145,135:POKE16081,160:GOTO420
370 POKE15506,157:POKE15507,180:POKE15508,144
371 POKE15572,131:POKE16082,159:POKE16083,129
372 POKE16019,184:POKE16020,135:POKE15956,160:GOTO420
373 POKE15573,157:POKE15574,180:POKE15957,159
374 POKE15958,129:POKE15894,184:GOTO420
375 POKE15639,151:POKE15640,141:POKE15895,135
376 POKE15831,181:POKE15832,158:GOTO420
377 REM --- DDRAW HALL TO THE LEFT
378 ONE+1GOTO379,391,394,396,398
379 GT=15369:GB=16265
380 GT=GT+1:GB=GB+1:IFGB=16270THEN420
390 POKEGT,176:POKEGB,140:GOTO380
391 POKE15502,151:POKE15503,131:POKE15504,131
392 POKE15505,131:POKE16078,181:POKE16079,176
393 POKE16080,176:POKE16081,176:GOTO420
394 POKE15570,151:POKE15571,131:POKE15572,131
395 POKE15954,181:POKE15955,176:POKE15956,176:GOTO420
396 POKE15573,181:POKE15574,176:POKE15893,157
397 POKE15894,140:GOTO420
398 POKE15639,157:POKE15640,140:POKE15831,151
399 POKE15832,131
415 REM --- IS THERE A WALL AHEAD?
420 IFQ=0THEN440
430 IF((X+(E*Y)=A(100))*(D=4))+((X+(E*Y)=A(101))*(D=2))
    THEN500
440 IFVAL(MID$(B$,3,1))=1THEN460
450 GOTO481
455 REM --- DRAW WALL AHEAD
460 ONE+1GOTO461,465,469,473,477
461 POKE16270,141:POKE16295,142:POKE15374,176
462 POKE15399,176:GT=15374:GB=16270
463 GT=GT+1:GB=GB+1:IFGT=15399THEN500
464 POKEGT,176:POKEGB,140:GOTO463
465 POKE15506,151:POKE15523,171:POKE16082,181
466 POKE16099,186:GT=15506:GB=16082
467 GT=GT+1:GB=GB+1:IFGT=15523THEN500
468 POKEGT,131:POKEGB,176:GOTO467
469 POKE15573,151:POKE15584,171:POKE15957,181
470 POKE15968,186:GT=15573:GB=15957
471 GT=GT+1:GB=GB+1:IFGT=15584THEN500
472 POKEGT,131:POKEGB,176:GOTO471
473 POKE15575,176:POKE15582,176:POKE15895,141
474 POKE15902,142:GT=15575:GB=15895
475 GT=GT+1:GB=GB+1:IFGT=15582THEN500
476 POKEGT,176:POKEGB,140:GOTO475
477 GT=15641:GB=15833
478 POKEGT,156:POKEGB,131
479 GT=GT+1:GB=GB+1:IFGT=15645POKE15644,172:GOTO500
480 POKEGT,140:POKEGB,131:GOTO479
481 E=E+1:IFE>4THEN500
483 GOSUB830
490 GOTO300
495 REM -- MOVE FORWARD, TURN RIGHT OR TURN LEFT
500 M$=INKEY$:IFM$=""THEN500
510 E=0:IFM$="F"THEN550
520 IFM$="R"THEN620
530 IFM$="L"THEN660
540 GOTO500
545 REM ---- MOVE FORWARD
550 CLS:IF(X=A(100))*(D=4)THEN700
560 IF(X=A(101))*(D=2)THEN760
570 GOSUB870
580 IFVAL(MID$(B$,3,1))=1THEN710
590 Q=Q+1:X=X+Y
600 GOSUB830
610 GOTO300
615 REM ---- TURN RIGHT
620 CLS:D=D+1:IFD<5THEN640
630 D=1
640 GOSUB830
650 GOTO300
655 REM ---- TURN LEFT
660 CLS:D=D-1:IFD>0THEN680
670 D=4
680 GOSUB830
690 GOTO300
695 REM -- WIN OR LOSE
700 PRINT@338,"YOU LOSE.  OUT AT ENTRANCE.":GOTO770
710 FORI=10TO19:FORJ=3TO4:SET(I,J+Z):NEXTJ:Z=Z+1:NEXTI
720 FORI=13TO8STEP-1:SET(19,I):NEXTI:Z=0
730 FORI=19TO33:FORJ=8TO9:SET(I,J+Z):NEXTJ:Z=Z+1:NEXTI
740 PRINT@530,"ZZZAAPPPP!!"
750 PRINT@653,"YOU JUST RAN INTO THE ELECTRIFIED WALL!"
752 GOTO770
760 PRINT@333,"YOU WIN IN";Q;"MOVES. ";A(102);
762 PRINT"IS MINIMUM SCORE."
770 PRINT:PRINT
772 PRINTTAB(13)"DO YOU WANT TO TRY AGAIN?  (Y=YES, N=NO)"
780 M$=INKEY$:IFM$=""THEN780
790 IFM$="Y"RESTORE:GOTO80
800 IFM$="N"THEN820
810 GOTO780
820 CLS:PRINT@320,"OK.  COME BACK WHEN YOU'RE":END
825 REM -- TEST FOR DIRECTION YOU ARE FACING
```

Program continues

✓See List of Advertisers on page 306

```
830 IFD=1THENY=-10
840 IFD=2THENY=1
850 IFD=3THENY=10
860 IFD=4THENY=-1
865 REM -- FETCH ARRAY A CONTENTS FOR CURRENT LOCATION
870 B$=STR$(A(X+(Y*E)))
875 REM -- ROTATE ARRAY A CONTENTS FOR CURRENT DIRECTION
880 IFD=1THEN902
890 FORI=2TOD
892 B4=VAL(MID$(B$,4,1)):B5=VAL(MID$(B$,5,1))
894 B6=VAL(MID$(B$,6,1)):B3=VAL(MID$(B$,3,1))
896 P=10000+B4*1000+B5*100+B6*10+B3
900 B$=STR$(P):NEXTI
902 RETURN
905 REM -- ARRAY D DATA
910 DATA1,8,7,14,2,8,8,20,3,7,10,23,4,10
912 DATA8,29,5,10,10,38,6,10,10,45
925 REM -- ARRAY A DATA FOR MAZE NR 1
930 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
940 DATA0,0,0,0,0,0,0,0,0,0
942 DATA11001,11010,11100,11011,11100,11011,11000,11100,0,0
950 DATA10011,11100,10111,11001,10010,11110,10101,10101,0,0
952 DATA11101,10101,11011,10100,11101,11001,10110,10001,0,0
960 DATA10000,10000,11100,10011,10100,10101,11011,10110,0,0
962 DATA10111,10101,10011,11000,10000,10010,11010,11100,0,0
970 DATA11001,10010,11100,10101,10011,11100,11101,10101,0,0
972 DATA10011,11010,10110,10011,11110,10011,10010,10110,0,0
980 DATA60,57,14,8,7
985 REM -- ARRAY A DATA FOR MAZE NR 2
990 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1000 DATA11101,11011,11000,11010,11000,11010,11100,11101,0,0
1002 DATA10011,11010,10000,11110,10011,11010,10100,10101,0,0
1010 DATA11001,11100,10001,11100,11001,11110,10001,10100,0,0
1012 DATA10101,10101,10111,10101,10101,11001,10110,10111,0,0
1020 DATA10000,10000,11010,10000,10100,10001,11010,11100,0,0
1022 DATA10101,10111,11101,10101,10101,10011,11100,10011,0,0
1030 DATA10001,11110,10001,10100,10011,11110,10101,11101,0,0
1032 DATA10011,11010,10110,10011,11010,11110,10011,10110,0,0
1040 DATA60,77,20,8,8
1045 REM -- ARRAY A DATA FOR MAZE NR 3
1050 DATA11001,11010,11100,11011,11000,11000,11100,0,0,0
1052 DATA10000,11100,10111,11101,10101,10101,10111,0,0,0
1060 DATA10101,10011,11000,10000,10110,10101,11101,0,0,0
1062 DATA10011,11100,10111,10111,11001,10010,10100,0,0,0
1070 DATA11101,10001,11000,11100,10101,11011,10110,0,0,0
1072 DATA10011,10110,10101,10111,10001,11110,11101,0,0,0
1080 DATA11001,11000,10110,11001,10010,11110,10101,0,0,0
1082 DATA10101,10101,11101,10101,11011,11000,10110,0,0,0
1090 DATA10001,10010,10100,10001,11010,10100,11001,0,0,0
1092 DATA10111,11011,10110,10011,11110,10011,10110,0,0,0
1100 DATA10,86,23,7,10
1105 REM -- ARRAY A DATA FOR MAZE NR 4
1110 DATA11011,11010,11000,11100,11001,11010,11000
1112 DATA11110,11011,11100,11001,11110,10101,10111
1114 DATA10011,11100,10101,11011,11100,10101
1120 DATA10101,11101,10011,11000,11010,10110,10101
1122 DATA11011,10100,10101,10100,10011,11000,10110
1124 DATA11001,11110,10001,11100,10001,10110
1130 DATA10001,11100,10001,11010,10100,11001,10100
1132 DATA10111,10001,11100,10101,10111,10101,11011
1134 DATA10110,10101,10101,11001,10110,10011
1140 DATA10001,11010,10110,11101,11101,10101,10001
1142 DATA10110,11101,11101,10011,11010,11010,10110
1144 DATA10011,10110,10011,11010,10010,10110
1145 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1150 DATA30,59,29,10,8
1155 REM -- ARRAY A DATA FOR MAZE NR 5
1160 DATA11001,11010,11100,11011,11100,11001,11110
1162 DATA11011,11100,11101,10100,11101,10001,11000
1164 DATA10110,10101,11001,11000,10010,10100
1170 DATA10101,10011,10110,10101,11101,10011,10100
1172 DATA10111,11101,10001,10001,11010,11000,10010
1174 DATA10110,11101,10011,11000,10110,10101
1180 DATA10011,11110,10101,11001,11010,10100,11101
1182 DATA10101,11011,10110,11001,11010,10100,10111
1184 DATA11011,10000,10110,10101,11001,11100
1190 DATA10101,11011,10110,11101,11101,10101,11011
1192 DATA10010,10100,10111,10001,11010,11000,10110
1194 DATA10011,10000,11000,11110,10001,11110
1200 DATA10101,11101,10101,11011,11100,10101,10101
1202 DATA11001,10000,11110,10011,10110,10011,11010
1204 DATA10010,10110,10011,10110,10011,11110
1210 DATA10,29,38,10,10
1215 REM -- ARRAY A DATA FOR MAZE NR 6
1220 DATA11001,11000,11010,11100,11001,11100,11011
1222 DATA11100,11011,11100,10111,10101,11101,10101
1224 DATA10111,10101,11001,10000,11010,10100
1230 DATA11000,10010,10110,10001,11000,10110,10101
1232 DATA10011,11110,10101,10101,11011,11000,10110
1234 DATA10011,11110,10011,11100,11001,10100
1240 DATA10011,11110,10101,11011,11000,11110,11101
1242 DATA10101,10111,10101,11001,11010,10000,11110
1244 DATA10101,11101,10011,10100,11101,10101
1250 DATA10011,11110,10101,11001,10100,10011,11010
1252 DATA10000,10110,10101,11001,11010,10100,10111
1254 DATA10001,11010,11010,10100,11001,10110
1260 DATA10011,11100,10011,11110,10101,11101,11101
1262 DATA10101,10011,11010,11011,10010,11010,11010
1264 DATA10010,10010,10110,10011,11010,11110
1270 DATA20,89,45,10,10
1280 END
```

Using the AY3-8910 with the TRS-80.

Polyphonic Sound Synthesis

Richard L. Brocaw
1850 Pearl Loop
Bosque Farms, NM 87068

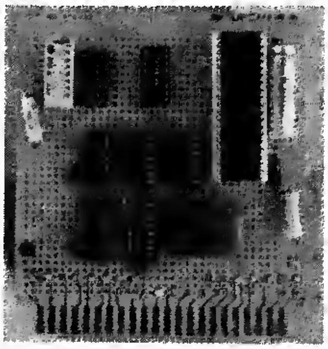


Photo 1. The complete sound generator board: At the upper left is the address selection switch, and at the lower right is the crystal and oscillator chip. Note the use of five bypass capacitors and one solid tantalum.

Being a fancier of game programs for the TRS-80, I recently became interested in complex sound generation, especially in the addition of exciting sound effects.

To produce the widest range of sound effects using the least amount of external circuitry, I narrowed my choice of sound chips to the General Instrument AY3-8910 Programmable Sound Generator (PSG). While this device is rather expensive ($15), its design and ease of use make it better than other chips.

The PSG is a bus oriented device and can be attached to almost any eight bit microprocessor bus. It is controlled by loading 14 registers on board the PSG. By selecting the correct combinations of registers under software control, any sound can be produced.

The PSG contains three independent tone generators with 16 levels of amplitude each; a variable noise generator; a set of mixers, and an envelope generator with eight selectable envelope shapes. In addition, the chip has two complete eight bit I/O ports which are addressable in a fashion similar to the 14 PSG registers. When using these ports, no other functions of the device are affected.

Sounds Good

Some combination of the 14 registers must be loaded with data in order to produce a sound. Each parameter must be analyzed into noise and/or tone components, envelope shape and speed. Once done, the registers can be loaded and the sound produced.

The first three register pairs are the tone period for each of the three channels. There is a coarse (four bit) and a fine (eight bit) register for each. Any tone frequency between about 111,500 Hz (000000000001) and 17.5 Hz (111111111111) may be produced. Each channel is independently programmable.

Channel	Coarse Register	Fine Register
A	R1	R0
B	R3	R2
C	R5	R4

Register R6 determines the frequency of the noise source in the PSG. Only the lower five bits are used. Noise frequencies between approximately 3,600 Hz (00011111) and 112,000 Hz (00000001) may be produced.

Register R7 controls the mixing of the three tone generators and the noise source. The two auxiliary I/O ports are controlled by R7. The table below shows the effect of each bit in R7:

BIT	=0	=1
7	I/O port B input	I/O port B output
6	I/O port A input	I/O port A output
5	Noise enable ch. C	Noise disable ch. C
4	Noise enable ch. B	Noise disable ch. B
3	Noise enable ch. A	Noise disable ch. A
2	Tone enable ch. C	Tone disable ch. C
1	Tone enable ch. B	Tone disable ch. B
0	Tone enable ch. A	Tone disable ch. A

Note that disabling noise and tone do not turn off a channel. This is done by writing zero to the channel's amplitude control register.

R8, R9, and R10 control the amplitude of channels A, B, and C. The lower four bits, B0-B3, allow 15 volume levels from mini-

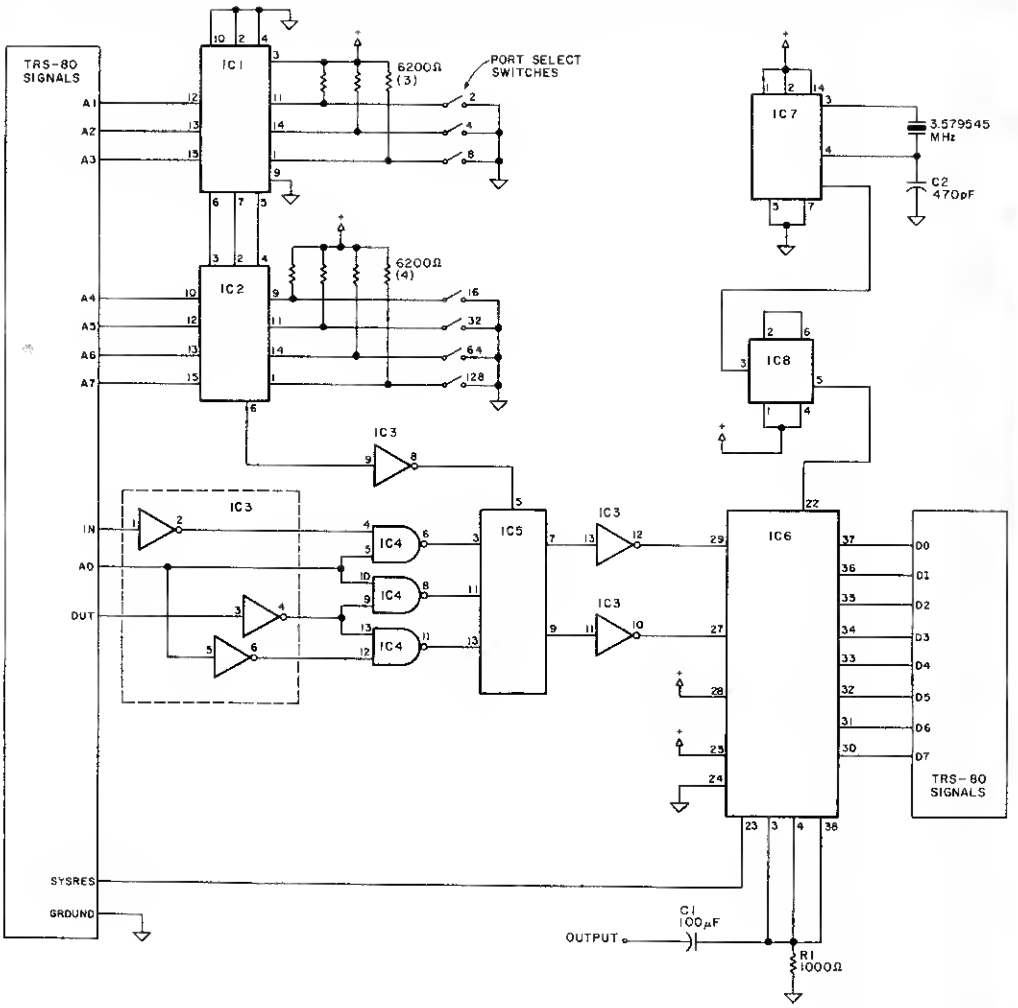


Fig. 1. PSG Plan

mum (0001) to maximum (1111). All zeros in the register turn off the corresponding channel. Bit B4 of each is the mode select. A one in this bit places the channel under control of the envelope generator. If bit B4 is one, no other bits have any effect.

The envelope period control is a 16 bit register made of registers R11 (fine period, lower eight bits) and R12 (coarse period, upper eight bits). The envelope period can be varied from approximately 0.11 Hz (00000000000000

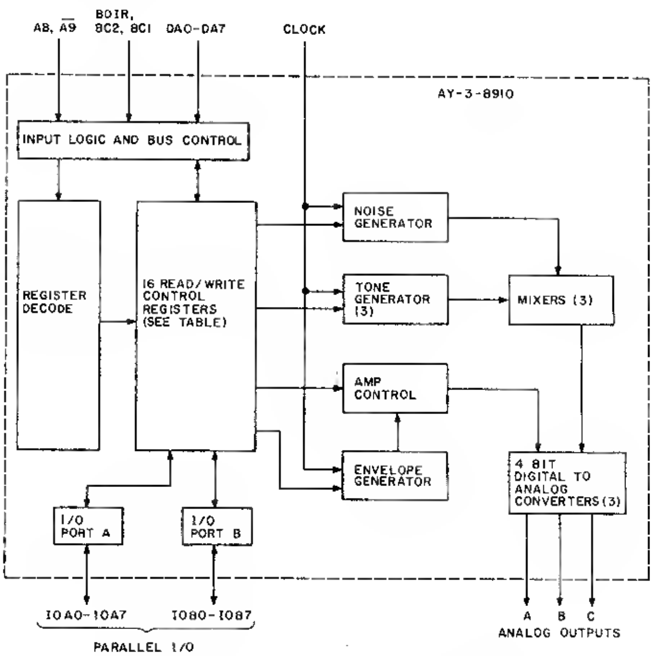


Fig. 2. Internal Organization of AY3-8910 PSG

CHIP	TYPE	POWER PIN	GROUND PIN
IC1	74LS85	16	8
IC2	74LS85	16	8
IC3	74LS04	14	7
IC4	74LS00	14	7
IC5	74148	16	8
IC6	AY3-8910	40	1
IC7	MC4024 VCO	14,1	5,7
IC8	74LS74	14	7

PSG Board Power requirement: +5 volts at .100 amps

Other Parts

C1 100 microfarad 15V electrolytic
C2 470 pf 5% silver mica
R1 1000 Ohm ½ watt

Table 1. TRS-80 PSG Parts List.

See List of Advertisers on page 306

SIGNAL	TRS-80 PIN
D0	30
D1	22
D2	32
D3	26
D4	18
D5	28
D6	24
D7	20
A0	25
A1	27
A2	40
A3	34
A4	31
A5	35
A6	38
A7	36
IN	19
OUT	12
SYSRES	2
Ground	8,29,37,39 (Connect TRS-80 grounds to PSG board ground and power supply ground)

Table 2. Connections to TRS-80 Expansion Port

01) to 6,990 Hz (1111111111111111).

The lower four bits of register R13 control the modulation pattern used on the mixer output, if the mode bit described earlier is a one. This register can be set up for single cycle, or repeating patterns. There are eight patterns described below:

BIT 3 2 1 0	ENVELOPE PATTERN
0 0 X X	Single cycle, start at maximum amplitude and decay to zero.
0 1 X X	Single cycle, attack from zero to maximum amplitude, then turn off.
1 0 0 0	Start at maximum amplitude and decay to zero, then repeat pattern.
1 0 1 0	Start at maximum amplitude, decay to zero, then attack to maximum. Repeat pattern.
1 0 1 1	Start at maximum amplitude, decay to zero, then hold at maximum amplitude.
1 1 0 0	Start at zero, attack to maximum amplitude, then turn off. Repeat pattern.
1 1 0 1	Start at zero, attack to maximum amplitude and hold at maximum.
1 1 1 0	Start at zero, attack to maximum amplitude, then decay to zero. Repeat pattern.

I/O Data Ports

Registers R14 and R15 are the auxiliary I/O ports. To output data from the bus to I/O port A, the following steps are required:

- Latch address R7
- Write a one to bit 6 of R7
- Latch address R14
- Write output data to port

The data output to the port will not change until a reset or another output comes along. Similar steps are required to input from the port.

This has been a brief overview of the PSG registers because a detailed account would cover many pages. The reader is urged to read the data manual before attempting any programming of the PSG. The manual's one defect, however, is its use of octal notation for registers and their contents. The description above

<table>
<tr><th>REGISTER</th><th></th><th>B7</th><th>B6</th><th>B5</th><th>B4</th><th>B3</th><th>B2</th><th>B1</th><th>B0</th></tr>
<tr><td>R0</td><td rowspan="2">CHANNEL A TONE PERIOD</td><td colspan="8">8 BIT FINE TUNE A</td></tr>
<tr><td>R1</td><td colspan="4"></td><td colspan="4">4 BIT COARSE TUNE A</td></tr>
<tr><td>R2</td><td rowspan="2">CHANNEL B TONE PERIOD</td><td colspan="8">8 BIT FINE TUNE B</td></tr>
<tr><td>R3</td><td colspan="4"></td><td colspan="4">4 BIT COARSE TUNE B</td></tr>
<tr><td>R4</td><td rowspan="2">CHANNEL C TONE PERIOD</td><td colspan="8">8 BIT FINE TUNE C</td></tr>
<tr><td>R5</td><td colspan="4"></td><td colspan="4">4 BIT COARSE TUNE C</td></tr>
<tr><td>R6</td><td>NOISE PERIOD</td><td colspan="3"></td><td colspan="5">5 BIT PERIOD CONTROL</td></tr>
<tr><td rowspan="2">R7</td><td rowspan="2">ENABLE</td><td colspan="2">IN/OUT</td><td colspan="3">NOISE</td><td colspan="3">TONE</td></tr>
<tr><td>IOB</td><td>IOA</td><td>C</td><td>B</td><td>A</td><td>C</td><td>B</td><td>A</td></tr>
<tr><td>R10</td><td>CHANNEL A AMPLITUDE</td><td colspan="3"></td><td>M</td><td>L3</td><td>L2</td><td>L1</td><td>L0</td></tr>
<tr><td>R11</td><td>CHANNEL B AMPLITUDE</td><td colspan="3"></td><td>M</td><td>L3</td><td>L2</td><td>L1</td><td>L0</td></tr>
<tr><td>R12</td><td>CHANNEL C AMPLITUDE</td><td colspan="3"></td><td>M</td><td>L3</td><td>L2</td><td>L1</td><td>L0</td></tr>
<tr><td>R13</td><td rowspan="2">ENVELOPE PERIOD</td><td colspan="8">8 BIT FINE TUNE E</td></tr>
<tr><td>R14</td><td colspan="8">8 BIT COARSE TUNE E</td></tr>
<tr><td>R15</td><td>ENVELOPE SHAPE CYCLE</td><td colspan="4"></td><td>CONT</td><td>ATT</td><td>ALT</td><td>HOLD</td></tr>
<tr><td>R16</td><td>I/O PORT A DATA STORE</td><td colspan="8">8 BIT PARALLEL I/O ON PORT A</td></tr>
<tr><td>R17</td><td>I/O PORT B DATA STORE</td><td colspan="8">8 BIT PARALLEL I/O PORT B</td></tr>
</table>

Fig. 3. Layout of registers: register numbers are in octal notation.

Photo 2. The complete sound generator system: The 5V power supply is at the upper right, the amplifier and speaker next to it. Three feet of twisted pair flat cable was used to connect the board to the TRS-80. The black wire in each pair is grounded for noise immunity. Note the use of the home brew connector for the TRS-80 at the lower left.

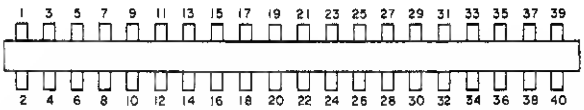


Fig. 4. Layout of bus edge connector on TRS-80: Keyboard viewed from the back with the edge connector on the right. Expansion interface viewed from the left side of the cabinet with the screen printer port to the right (line printer port to the left).

is based on the decimal equivalents.

TRS-80 Adaptation

On obtaining an AY3-8910, I found no data was available explaining how to interface the device to the TRS-80. A diagram was supplied for the S-100 bus. Using that and the detailed technical write up, I designed the following interface.

The circuitry required is simple and straightforward. Referring to the schematic, IC1 and IC2 select the lower of the two required consecutive address ports. The lower port address is the sum of the open switches. Some of this circuitry may be eliminated by hardwiring the port address and leaving out the switches and pullup resistors. Outputting to the first port causes the register address to be latched in the PSG. Outputting to the second port loads the selected register with data. An input from the second reads the data in the selected register. The priority encoder, IC5, in combination with the in and out signals from the TRS-80, simulates the bus control signals required to control the operation of the PSG. IC7 and IC8 provide the clock signal.

The choice of the MC4024 voltage controlled oscillator for IC7 was arbitrary; any TTL oscillator would work. The crystal frequency is not critical, but for correct tone frequencies, a TV color burst crystal is recommended.

My breadboard version worked perfectly with a 4.7 MHz crystal, although the tone frequencies were high.

The PSG IC6 has the three outputs tied together. The output is suitable for driving any standard audio amplifier. A stereo effect could be produced by using separate amplifiers in each output, but the R1, C1 network must be repeated at each output. Note that no bi-directional drivers are required on the data bus; the PSG has tri-state drivers built in for these lines.

The circuit was assembled on a Radio Shack kluge card using wirewrap. It was connected to the TRS-80 via three feet of twisted pair flat cable. Parts lay-

✓See List of Advertisers on page 306

out is not critical. The board should have the usual complement of bypass capacitors. Any power supply capable of +5 volts at 0.100 amp should be adequate. Be sure that the ground pins on the TRS-80, the ground on the PSG board, and the power supply ground are all tied together.

I had a problem in the breadboard version when using a prewired 40-conductor edge connector because of noise on the unterminated lines. I recommend that only the required signals be cabled from the TRS-80. A suitable edge connector can be made by hacksawing the first 20 pins from the end of an S-100 edge connector.

The circuit was designed to plug into the screen printer port on the expansion interface. If the reader wishes to hook up directly to the TRS-80 keyboard, additional buffering of all lines, except of the data bus, may be required. This prevents overloading the signals.

Programming

When ordering an AY3–8910, the purchase of the complete data manual is recommended because the methods of calculating tone and noise frequencies are not included in the abbreviated data sheet.

While programming the PSG is beyond the scope of this article (the subject occupies about 10 pages in the manual), the following routines are presented. They show the structure of PSG programs. The address latch port is 16 and the read/write port is 17.

Routine 1: Gunshot

```
10 'RESET ALL PSG REGISTERS
20 FOR I = 1 TO 14: OUT 16,I: OUT 17,0:
   NEXT I
30 OUT 16,6: OUT 17,15 'SET NOISE
   PERIOD TO MID VALUE
40 OUT 16,7: OUT 17,7 'ENABLE NOISE
   ON CHANNELS A,B,C
50 OUT 16,8: OUT 17,16
60 OUT 16,9: OUT 17,16 'SELECT FULL
   AMPLITUDE RANGE UNDER
70 OUT 16,10: OUT 17,16 'CONTROL OF
   THE ENVELOPE GENERATOR
80 OUT 16,12: OUT 17,16 'SET ENVE-
   LOPE PERIOD TO .586 SECONDS
90 OUT 16,13: OUT 17,0 'SELECT ENVE-
   LOPE DECAY, ONE CYCLE
100 FOR I = 1 to 100: NEXT I 'DELAY TO
   NEXT GUNSHOT
110 GOTO 90
```

Note: Once the registers are set up, the sound effect is repeated by triggering the envelope cycle, statement 90. By stretching the envelope period to about two seconds at statement 80, and using a longer delay at statement 100, the sound effect becomes an explosion.

Routine 2: Siren

```
10 'RESET ALL PSG REGISTERS
20 FOR I = 1 TO 14: OUT 16,I: OUT 17,0:
   NEXT I
30 OUT 16,7: OUT 17,62 'ENABLE TONE
   ON CHANNEL A
40 OUT 16,8: OUT 17,15 'SELECT MAXI-
   MUM AMPLITUDE ON CHANNEL A
50 OUT 16,0 'SELECT CHANNEL A TONE
   REGISTER
60 FOR J = 1 TO 3 'PRODUCE 3 CYCLES
   OF THE SIREN
70 FOR I = 100 TO 200 'DOWNWARD
   FREQUENCY SWEEP
80 OUT 17,I
90 NEXT I
100 FOR I = 200 TO 100 STEP -1 'UP-
   WARD FREQUENCY SWEEP
110 OUT 17,I
120 NEXT I
130 NEXT J
140 OUT 16,8: OUT 17,0 'SET AMPLITUDE
   TO ZERO TO END EFFECT
```

Note: The tone is turned on at statement 40 and off at 140. Breaking will allow the tone at the selected frequency to remain constant. Once the register address is latched, the register can be read or written many times without reselecting the register.

These are only two of the audio programs that could be written for the PSG. In addition, the PSG could be used as a music synthesizer for the creation of two and three part harmony. This author has written a series of programs in BASIC, giving further examples of PSG usage, and a utility program which allows calculation of the contents of the registers. It also examines the current status of PSG registers.

Once the reader has built and installed the circuit, he should have hours of fun experimenting with myriad effects the device produces. I found the total cost to be about $30 using new parts. The entertainment value of the games and the challenge more than offset the cost.

The AY3–8910 is available off the shelf from many companies advertising in *80 Microcomputing.* ■

Note: These programs are available from the author.

GRAPHIC DIVERSIONS!

3-D Adventures

Our 3-D adventures combine the traditional word oriented adventure with full screen 3-D perspective graphics! Hallways, stairs, elevators, objects, and even characters in Asylum, are represented in 3-D graphics *as though you are actually there!*

"... truly one of the best adventure games by any criteria... it borders on the unbelievable. Believe it. Buy it." M.J.M., *The Space Gamer*

"This has got to be the most infuriating, irritating, aggravating, frustrating, angering, spellbinding game on the market." D.M., *'80 Microcomputing*

ASYLUM places you in the residence home for Deathmaze survivors. To leave, you must deal with guards, fellow survivors, doctors, the infamous crazed carpenter, and much, much more.

TRS-80 Level II 16K or Model III 16K	$14.95
Model I DISK	$19.95

LABYRINTH places you in a huge maze of tunnels inhabited by gnomes, ghosts, witches, and an evil minotaur. You must find the weapons and treasures needed to destroy the minotaur before he destroys you!

TRS-80 Level II 16K or Model III 16K	$12.95
Model I DISK	$17.95

DEATHMAZE 5000 places you in a gigantic five-story building. There is only one goal. ESCAPE ALIVE! Monsters, dogs, vampires, and other vile horrors will plague your every step as you struggle to survive one of the most challenging adventures ever written.

TRS-80 Level II 16K or Model III 16K	$12.95
Model I DISK	$17.95

GRBASIC

GRBASIC extends Level II or DISK BASIC to include an easy to use graphics command set. A single BASIC command allows the user to draw a line between any two pixels on the screen in hundredths of a second! Coordinates can be chained to allow complex figures to be drawn by a single BASIC program line in less than a second!

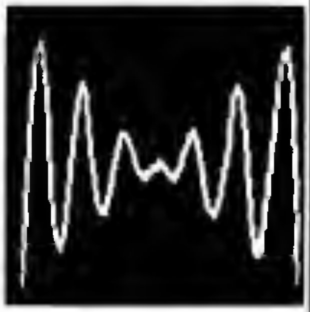
GRBASIC allows the definition of shapes. Once defined, a shape can be **rotated**, **scaled** up or down in size, drawn anywhere on the screen in less than a second, and can even be drawn totally or partially "off" the screen in extended space! And all with short, simple BASIC commands! Even multiple shapes are no problem!

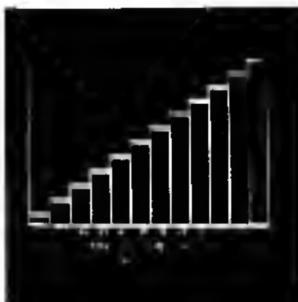
GRBASIC is not a string-packer or machine-language USR-called utility. GRBASIC is fully integrated into Level II and DISK BASIC. There is nothing on the market that offers its graphic features, except possibly the Apple II computer! Animation, scientific plotting, and professional data displays are now child's play!

A new program, GRBASIC FUNCTION PLOTTER, allows the plotting of almost any function, including polar coordinate based figures and almost any wave form. Features include function definition and automatic screen scaling. REQUIRES GRBASIC!

GRBASIC cassette	$19.95
GRBASIC TRS-DOS disk	$24.95
GRBASIC NEWDOS80 disk	$24.95
GRBASIC FUNCTION PLOTTER	$14.95

Examples of graphics produced with GRBASIC





Med Systems Software
P.O. Box 2674-S Chapel Hill, NC 27514
(919) 933-1990

MICRO-WORLD

JOURNEY TO THE CENTER OF THE TRS-80!

Micro-World is an adventure like no other. You are transformed into an electroid, and must travel through the circuits of your TRS-80! You will be required to solve many incredible problems, and in the process you will gain a knowledge of how a TRS-80 operates.

Micro-World is one of the most advanced pieces of machine language programming Med Systems has published. A special encoding scheme has allowed a 21K adventure to fit in 16K. Micro-World is verbose. Messages are frequent and fact filled. There are over 80 locations that must be explored.

Micro-World is an excellent educational simulation! It is supplied with a booklet containing a glossary and explanations of the electronic circuits inside the TRS-80. This does not in any way diminish the challenge! If anything, the challenge increases since you must gain a working knowledge of your computer to gain access to the final circuit!

Model I Level II 16K or Model III 16K cassette	$19.95
Model I disk	$24.95

The Playful Professor

The Playful Professor has been a constant best-seller since its introduction in 1980. This program is a mathematical tutorial that provides instruction in addition, subtraction, multiplication, and division, with or without fractions. Problems are presented in a game format that gives a step-by-step tutorial for incorrect answers.

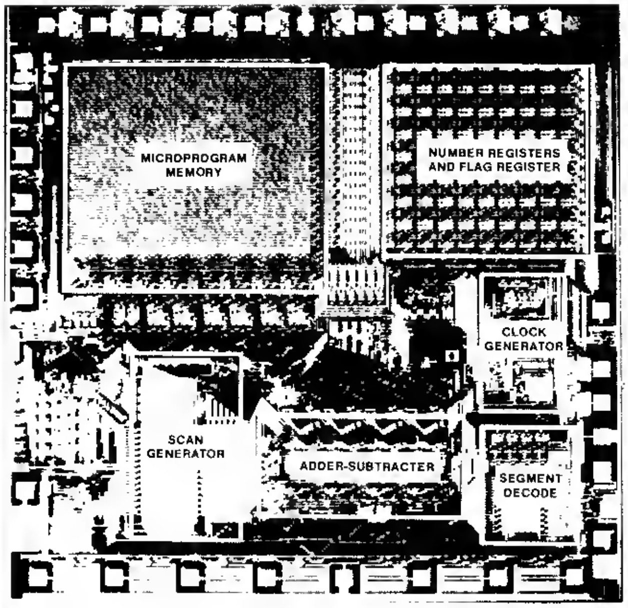
The Playful Professor places the user in a 60 room mansion haunted by an intelligent ghost who holds the key to the only door out. Options include 3 difficulty levels, choice of problem type, 1 or 2 players, pass option, and split difficulty levels to allow 7 year olds to play competitively with 27 year olds! Graphics are used extensively.

"I'm so impressed with The Playful Professor, Money Master, and Deathmaze. At the Mead School we use these three programs daily." R.J., Greenwich, Conn.

"My 7 year old daughter had a fun way to review her math . . . my 4 year old enjoys playing using the pass feature (what an experience it is to watch a 4 year old use a computer!)" S.C., Highland, Maryland

Model I or Model III Level II 16K	$12.95
Model I DISK	$16.95

Med Systems Software
P.O. Box 2674-S Chapel Hill, NC 27514
(919) 933-1990



SATISFACTION GUARANTEED!

If for any reason you are not satisfied with our products, return your order within 14 days for a prompt and cheerful refund.

ORDERING INFORMATION

Orders are processed within five working days. We pay all postage and handling within the U.S., Canada, and U.S. territories. European orders please include $3.00 for air post.

☐ Asylum	($14.95)	$ ______
☐ Deathmaze 500	($12.95)	$ ______
☐ Labyrinth	($12.95)	$ ______
☐ Adventures on Disk	(add $5.00)	$ ______
☐ GRBASIC Cassette	($19.95)	$ ______
☐ GRBASIC TRS-DOS Disk	($24.95)	$ ______
☐ GRBASIC NEWDOS 80 Disk	($24.95)	$ ______
☐ GRBASIC FUNCTION PLOTTER	($14.95)	$ ______
☐ Playful Professor	($12.95)	$ ______
☐ MICRO-WORLD	($19.95)	$ ______
☐ MICRO-WORLD Disk	($24.95)	$ ______
	Total	$ ______

Name ______________________

Street ______________________

City __________ *State* __________ *Zip* ______

Computer:

☐ TRS-80 LII 16K ☐ Model III 16K

☐ *Mastercard* ☐ *Visa* ☐ *check*

MC or VISA # ______________________

Expiration Date ______________________

✓128

A quick look at an alternative to BASIC.

The Pascal Dream

John Krutch
P.O. Box 761
Crescent City, CA 95531

Just three years ago, when the TRS-80 was introduced, the only language available was Level I BASIC. Not even T-BUG or the Microsoft editor/assembler had been released.

But things have changed in three short years. TRS-80 owners can now pick and choose from many languages. A few of them are supplied by Radio Shack, but many of them come from other vendors. Besides the BASIC interpreters and compilers, there's BASEX (a structured language similar to BASIC), Pascal, FORTRAN, COBOL, Tiny C, FORTH, APL, LISP, PILOT, and others. Most are available for both cassette and disk-based systems. And if you're fortunate enough to have a modified TRS-80 that runs standard CP/M by Digital Research, your choice is even greater.

I want to discuss an interesting Pascal compiler, the Pipe Dream Pascal system (better known in the United States as People's Pascal I). I want to demonstrate a feature of the Pipe Dream system that sets it apart from most other Pascal compilers on the market.

Pascal Compiler-Interpreter

What is a compiler, anyway? A compiler is a piece of software that takes a source program—that is, a program written in a high-level language—and translates it into a low-level object program equivalent to the source program.

Theoretically, then, if you have a compiler for language X running on your TRS-80, you ought to be able to type in a source program written in X; the compiler then compiles it into fast, compact Z-80 object code. Then, next time you want to run the program, you can just load the object file; the source code may be thrown away. David Bolke's TINYCOMP (*80 Microcomputing*, May) takes a BASIC source file and compiles it into pure binary machine code.

Unfortunately, few compilers for microcomputers work this

```
10     (* Peeker *)
20
30     VAR   ADDRESS, BYTE : INTEGER;
40
50     BEGIN
60       REPEAT
70         WRITE('ADDRESS? ');
80         READ(ADDRESS#);
90         BYTE := MEM(ADDRESS);
100        WRITE('CONTENTS = ', BYTE#, 13, 13)
110      UNTIL 1 <> 1
120    END.
```

Program Listing 1

```
10     (* Peeker *)
20
30     VAR   ADDRESS, BYTE : INTEGER;
                                                  0  JMP  0  0
40
50     BEGIN
60       REPEAT
                                                ADD 1 AT 0
                                                  1  INT  0  5
70           WRITE('ADDRESS? ');
                                                  2  LIT  0  65
                                                  3  LIT  0  68
                                                  4  LIT  0  68
                                                  5  LIT  0  82
                                                  6  LIT  0  69
                                                  7  LIT  0  83
                                                  8  LIT  0  83
                                                  9  LIT  0  63
                                                  10  LIT  0  32
                                                  11  LIT  0  9
                                                  12  CSP  0  8
80           READ(ADDRESS#);
                                                  13  CSP  0  2
                                                  14  STO  0  3
90           BYTE := MEM(ADDRESS);
                                                  15  LOD  0  3
                                                  16  LOD  255  0
                                                  17  STO  0  4
100          WRITE('CONTENTS = ', BYTE#, 13, 13)
                                                  18  LIT  0  67
                                                  19  LIT  0  79
                                                  20  LIT  0  78
                                                  21  LIT  0  84
                                                  22  LIT  0  69
```

Program continues

See List of Advertisers on page 306

```
                                        23  LIT  0  78
                                        24  LIT  0  84
                                        25  LIT  0  83
                                        26  LIT  0  32
                                        27  LIT  0  61
                                        28  LIT  0  32
                                        29  LIT  0  11
                                        30  CSP  0  8
                                        31  LOD  0  4
                                        32  CSP  0  3
                                        33  LIT  0  13
                                        34  CSP  0  1
                                        35  LIT  0  13
                                        36  CSP  0  1
110      UNTIL 1 <> 1
                                        37  LIT  0  1
                                        38  LIT  0  1

120    END.
                                        39  OPR  0  9
                                        40  JPC  0  2
                                        41  OPR  0  0
```

Program Listing 2

straightforwardly. All compilers take a source file and translate it into an object file, however, the object code produced in this way isn't necessarily code that can be read and executed by the host computer's processor.

Let's look at UCSD Pascal, which is available for many microcomputers (including the Model I and Model II TRS-80). UCSD Pascal consists of an interpreter and a compiler. When the UCSD system running on a TRS-80 compiles a Pascal source file into object code, what is produced is actually a form of intermediate code known as *pseudocode* or *p-code*. This p-code then goes to the interpreter which executes the p-code, causing the program to run. This is as far as you can go with UCSD Pascal.

Pascal Translator

Pipe Dream Pascal is a "Tiny" Pascal designed along the lines suggested by Kin-man Chung and Herbert Yuen in 1978. It was written by John Alexander of Berwick, Australia and is distributed in this country by Computer Information Exchange, Inc. as People's Pascal I. It is available on cassette for the 16K Level II TRS-80.

Pipe Dream Pascal includes a compiler which compiles Tiny Pascal source into p-code, and an interpreter which executes the p-code. Pipe Dream Pascal also supplies a translator, which does just what its name suggests—it takes the p-code generated by the compiler and translates it into pure binary Z-code.

Using the Pipe Dream system, therefore, can write your Tiny Pascal program, compile it, and execute the resulting p-codes with the interpreter. Once you've executed the program and verified that it is operating correctly, you can use the translator to generate Z-80 machine code from the p-code. The resulting Z-code program is fast and occupies very little memory. (However, you'll probably need to load a run-time system along with the Z-code program to handle I/O operations and such. This system occupies only about 1K bytes.)

Program Listing 1 is a short program written in Pipe Dream's subset of Pascal, which prompts the user to type in a memory address in decimal, and returns the byte at that location. For readers not familiar with Pascal, here a few words of explanation. Line 30 sets variables ADDRESS and BYTE to integer type.

Lines 60-110 set up a loop which repeats indefinitely, since the condition in line 110 (1<>1) can never be met. Line 70 prompts the user for the address to be PEEKed at. Line 80 reads this address from the keyboard into variable ADDRESS. Line 90 gets the byte

```
        1.                 2.     3.    4.
                           23000  59D8  DD21D759DDF921FA49E5E5E5
 0 JMP 0   1
 1 INT 0   5               23012  59E4  21F6FF39F9
 2 LIT 0   65              23017  59E9  E741
 3 LIT 0   68              23019  59EB  E744
 4 LIT 0   68              23021  59ED  E744
 5 LIT 0   82              23023  59EF  E752
 6 LIT 0   69              23025  59F1  E745
 7 LIT 0   83              23027  59F3  E753
 8 LIT 0   83              23029  59F5  E753
 9 LIT 0   63              23031  59F7  E73F
10 LIT 0   32              23033  59F9  E720
11 LIT 0   9               23035  59FB  E709
12 CSP 0   8               23017  59E9  CDB849414444524553533F2000
13 CSP 0   2               23030  59F6  CDA649
14 STO 0   3               23033  59F9  F7FA
15 LOD 0   3               23035  59FB  EFFA
16 LOD 255   0             23037  59FD  CDF749
17 STO 0   4               23040  5A00  F7F8
18 LIT 0   67              23042  5A02  E743
19 LIT 0   79              23044  5A04  E74F
20 LIT 0   78              23046  5A06  E74E
21 LIT 0   84              23048  5A08  E754
22 LIT 0   69              23050  5A0A  E745
23 LIT 0   78              23052  5A0C  E74E
24 LIT 0   84              23054  5A0E  E754
25 LIT 0   83              23056  5A10  E753
26 LIT 0   32              23058  5A12  E720
27 LIT 0   61              23060  5A14  E73D
28 LIT 0   32              23062  5A16  E720
29 LIT 0   11              23064  5A18  E70B
30 CSP 0   8               23042  5A02  CDB849434F4E54454E5453203D2000
31 LOD 0   4               23057  5A11  EFF8
32 CSP 0   3               23059  5A13  CDA949
33 LIT 0   13              23062  5A16  E70D
34 CSP 0   1               23064  5A18  CDA349
35 LIT 0   13              23067  5A1B  E70D
36 CSP 0   1               23069  5A1D  CDA349
37 LIT 0   1               23072  5A20  E701
38 LIT 0   1               23074  5A22  E701
39 OPR 0   9               23076  5A24  CDD949
40 JPC 0   2               23079  5A27  F1D2E959
41 OPR 0   0               23083  5A2B  CDBE49
                           23086  5A2E  C30000
```

Program Listing 3

out of memory and puts it into variable **BYTE**. Line 100 writes the byte on the display and prints two line feeds.

Program Listing 2 shows the same program and its compilation into p-code. Each group of p-codes is listed just underneath the line of source code. The number that precedes the p-code is the location number of the p-code. There are 42 p-codes numbered 0-41. The two numbers that follow each p-code, are operands.

As to what the p-codes do, look at p-code number 2, LIT 0 65, which causes the number 65 to be loaded on top of the stack. This number is the decimal ASCII code for A, which is the first letter of the word ADDRESS?. This will be printed on the display to prompt the user. P-code number 34, CSP 0 1, is a call to standard procedure 1. This causes a jump to subroutine 1 in the run-time system.

Program Listing 3 is the culmination of our work and is the output from the translator, which decodes the p-code into Z-code. Column one is the list of p-codes. Column two contains the decimal address at which the translated Z-80 codes begin for each p-code. Column three contains the same address in hexadecimal. Column four is the actual machine code. It is not an *exact* translation into machine language of the source program in Program Listing 1, because it depends on a run-time system which must be loaded when the machine code is.

The Pipe Dream Pascal system is cumbersome. First, you have to load the compiler, then the interpreter, then the translator, and finally the run-time system. Then you have to load T-BUG to run off a copy of the translated program and run-time system. However, in spite of all this cassette-swapping, using the Pipe Dream system is a fascinating experience. It is rewarding to watch a program written in a *very* high-level language turn into binary machine code right before your eyes.■

✓See List of Advertisers on page 306

Why sit around typing RUN and LIST all day?

The Disk Auto-Menu

Larry M. Hewin, Ed. D.
Volksmicro Computer Systems, Inc.
202 Packets Court
Williamsburg, VA 23185

1	NAME FILE	2	DISK DUMP UTILITY
3	DISPLAY DOS LIB CMDS	4	DIR DRIVE 0
5	DISPLAY FREE DISK SPACE	6	EXPENSE RECORD
7	APPOINTMENTS	8	ANOTHER DISK MENU
9	MENU MAINTENANCE	10	PRINTED MENU

ENTER SELECTION = =>?_

Table 1

While showing off my microcomputers to customers and clients, I've frequently found myself typing RUN WHATEVER/BAS and other file names again and again. To make it harder, my typing error rate goes up as I hurry to lessen the dead time during loading. As a result, "the hurrier I go, the behinder I get."

I've tried setting up demo disks in advance. The items that I desire to show vary, however, so preparing a set of demos doesn't satisfy my need.

Besides, all of us know the one thing a computer is supposed to do well is take away the need for humans to perform repetitious and dull tasks.

So why sit around typing RUN or LOAD such and such on a machine which is capable of doing it better? The two simple programs in this article reflect an approach to a solution.

Main Point

The main objective is to choose a program from a menu that tells which programs are residing on the diskette(s) in the drives, and to call the selection by entering the number of the choice. A second objective is to be able to return from the selection or called program to the disk menu, to call another. Easy.

Also, since I'm not always certain what some of my more cryptic file names represent, it would be nice to have the menus present descriptive titles. This would identify, while still allowing the use of, the actual filespec for calling.

It seems desirable to provide for a printed version of diskette screen menus, since the menu collection itself provides only a primitive directory of the program collection.

Finally, inherent in the collective points above is the need for a data base manager with the usual entry, edit and delete functions.

Two Programs

The system devised consists of two main programs: a menu maintenance program and a menu manager program. The menu maintenance program allows entry, editing and deletion of program descriptions; and calling filespecs. Under the file name MENU/DAT, it writes data to the disk you have designated.

MENU MAINTENANCE FUNCTIONS	
NEW MENU	1
EDIT MENU	2
DELETIONS	3
ADD PROGRAMS	4
RUN MENUMGR	5

ENTER SELECTION = =>?_

Table 2

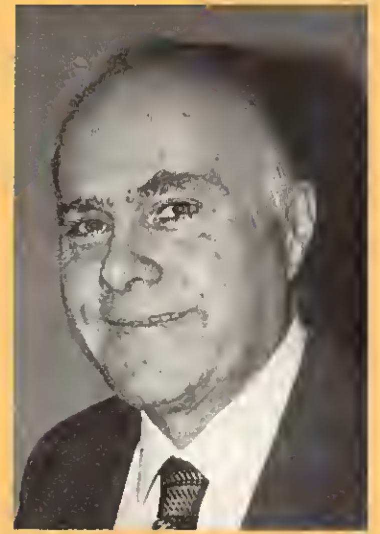
The menu manager program, when run, reads data from MENU/DAT for whichever disk you've designated and screens it in menu format (Table 1). You are then free to select a program by number from the menu and call it by entering its number. You may also call the following items from the menu: a different disk drive menu; the menu maintenance program; or a return to DOS. If you are in the menu maintenance program for any reason, the menu there permits you to call the menu manager program.

The two main programs are saved to a disk containing a DOS you normally have in drive 0. These two need to be in your drives on one disk only, so it may make sense to put them on the DOS disk. MENU/DAT will be on every disk used in the system and contains only the menu data for that diskette.

To start using this system, you load BASIC, and run MENUMTC/BAS. You are then asked to enter the drive number on which the data is to be entered.

From the menu of menu maintenance functions as shown in Table 2, select New Menu by entering 1. You will then be given prompts for entering program data which you answer in the manner shown in Table 3. When the two fields are entered, the

"I'm Wayne Green and I can save you $986 on the purchase of a computer system!



I hope that has your attention."

"That's the difference between walking into your local Radio Shack store and plunking down hard cash . . . and buying from the ads in 80 Microcomputing magazine. That's the difference for a simple combination such as a Model III two disk system with 48K of memory, a modem and a Line Printer II: The Radio Shack price for that combination is $3,612. If you buy from the ads in my magazine, you'll buy exactly the same system for $2,626. So why throw away $986?

"The fact is that the money you can save on even the smallest accessory purchase will pay for the magazine subscription many times over. That's one of the reasons so many people are subscribing to 80.

"Another is that it is the major source of information on the TRS-80 computer. In 1980 there were 335 feature articles on the system . . . with detailed instruction on how to do things (sorry about that), evaluations of accessories and software . . . and so on. I guarantee you'll find the magazine invaluable.

"A subscription to 80 is still only $18 (when are we going to raise that darned price to $25, where it should be?), so get your subscription in before I boost that price. It could be any day now."

Wayne Creen
Editor/Publisher

If the coupon below has been used, please fill out subscription form on the Reader Service card in the back of the magazine

YES! Sign me on as a subscriber to 80 Microcomputing for only $18 a year!

Card #__________ Exp.______

Signature __________

Name __________

Address __________

City ________ State ______ Zip ______

- ☐ 12 issues – $18
- ☐ 36 issues – $45
- ☐ Please bill me
- ☐ Payment Enclosed
- ☐ Master Charge
- ☐ VISA
- ☐ American Express

80 microcomputing
Peterborough, N.H. 03458

Subscription begins with next published issue
Back issues, while available are $3 each
Canada $20 per year US funds
All other foreign subscriptions $28 one year only

316B8

program automatically asks whether the entries are correct or not. If the answer is N for no, the program cycles you back through the entries. If a particular entry is correct, hit ENTER and the entry, as originally entered, will be preserved. If it isn't correct, however, enter the correct one.

While BASIC programs are entered as shown in Table 3, command programs, usually summoned from DOS, need different treatment. In the program title field we can still name programs anything we desire, but in the field for filespec the procedure is as follows: Assume the program is a machine language program called Electric Stylus and has the filespec: STYLUS/CMD. The proper entry to the prompt for filespec is &STYLUS, i.e., an ampersand is put in front, and the / and file extension is left off.

Providing for an automatic return to DOS is similar. If you respond with RETURN TO DOS to the program title prompt, and respond to the filespec prompt with &DIR, the menu will show an item returning you to DOS, and call up a DIR command. Similar entries, &LIB and &FREE, will invoke those DOS commands.

Exit

When you are finished entering programs, the exit is accomplished by answering the program title prompt with END. The program then leads you to save your entries on disk. It specifies the disk drive number on which they are to be saved. It then returns to the menu of the menu maintenance program. You may then elect to run the menu manager program by making the appropriate selection, and when prompted, giving the disk drive number for the disk on which you have just saved your MENU/DAT. This brings up the menu you have established, and allows menu selection of programs. Note that the program has automatically added the selections necessary for calling a new drive menu, menu maintenance, and for a printed copy of the menu.

How to return from called programs to the menu? The answer to this problem depends on whether it is a BASIC program or machine language or other entry which calls for exiting the BASIC mode.

The approach with BASIC programs is simple. I usually modify one line of the called program; where a program line might have been 900 END, I substitute 900 RUN MENUMGR/BAS. If the called program has a menu, I usually modify it to add on an item like RETURN TO DISK (or MAIN) MENU (9). I modify the program so that the (9) selection sends it to a line like the line 900 above.

With machine language programs, the problem is different. It is generally difficult to modify programs to call MENUMGR/BAS directly (particularly if you did not write them, or have not disassembled them). There are, however, possibilities in short machine language instructions, designed to bring the MENUMGR/BAS back in when the reset button is hit. This is accomplished by Autoing the machine language program.

The approach I have chosen was selected because it has been printed in *80 Microcomputing* (May, 1980, page 27, in an excellent letter from Gary Alcorn,

MENU ENTRIES

FOR PROGRAM # 1 :
THE PROGRAM TITLE IS: NAME FILE
THE FILESPEC IS: NAMEFILE/BAS
ARE THESE CORRECT - Y/N = =>?__

Table 3

The System is comprised of three programs:
1. MENUMTC/BAS—the data manager for entry and maintenance of menu data for each of the user's disks. Writes data to the disk under the filespec MENU/DAT.
2. MENUMGR/BAS—the program that calls MENU/DAT, writes the screen menu, and organizes program selection from the menu.
3. MENU/CMD—a machine language program which, when AUTOed, allow return to MENUMGR/BAS anytime reset is hit.

Fig. 1

1. Enter TRSDOS 'DEBUG' and type D6B00
2. Type M6B00(space)
3. Enter machine code exactly as shown:

```
6B00 21 0F 6B 22 16 40 21 2C 6B 22 2A 6B C3 2D 40 E5
6B10 2A 2A 6B 7E FE 0A CA 1F 6B 23 22 2A 6B E1 C9 21
6B20 E3 03 22 16 40 3E 0D C3 1D 6B 20 20 1F 42 41 53
6B30 49 43 0D 0D 0D 52 55 4E 22 4D 45 4E 55 4D 47 52
6B40 2F 42 41 53 22 0A
```

4. Press Enter
5. Type G402D then hit Enter
6. In DOS mode type DEBUG (OFF)
7. Type TAPEDISK and hit Enter
8. Save program with TAPEDISK by
   ?F MENU/CMD:0 6B00 6B45 6B00
   ?E (enter) to return to DOS READY

Example 1

Program Listing 1

```
1 REM    * COPYRIGHT DR. LARRY HEWIN
2 REM    * VOLKSMICRO COMPUTER SYSTEMS INC.
3 REM    * 202 PACKETS COURT
4 REM    * WILLIAMSBURG, VA 23185
5 REM    * DECEMBER 15, 1979
6 REM    ********************************
7 REM    * A DISK MENU SYSTEM -PRGRM # 1 *
8 REM    * MENU MANAGER (FS=MENUMGR/BAS) *
9 REM    ********************************
10 CLEAR5000:CLS:DEFINT C,I,L,N,P,X:DIMF$(26),FS$(26):U
     $="==>":U1$="##"
11 GOTO40
15 INPUT"PRESS ENTER TO CONTINUE";Z$:RETURN
30 FORI=1TO1000:NEXT:RETURN
40 CLS:PRINT@474,"MENU MANAGER":PRINT@597,"BY DR. LARRY
      M. HEWIN"
50 PRINT@780,"ENTER DRIVE NO. FROM WHICH MENU IS DESIRE
     D"U$;:INPUTD$
60 IFVAL(D$)<0ORVAL(D$)>4THENPRINT"PLS REDO":GOTO50
100 S$="MENU/DAT:"+D$:OPEN "I",1,S$
110 CLS:PRINT@464,"LOADING DATA FROM DRIVE NO. "D$:INPU
     T#1,N
120 FORI=1TON:INPUT#1,F$(I):LINEINPUT#1,FS$(I):PRINT@60
     0,"RECORD NO. "I;:NEXTI:CLOSE:CLS:PRINT@477,"DONE"
     :GOSUB30:CLS
200 PRINT@30,"MENU":L=2:C=1:X=1:P=0
210 PRINT@(L*64+X),USINGU1$;C;:PRINT"   ";F$(C)
220 C=C+1:IFC>NTHENGOSUB300:GOTO260
230 X=X+32:IFX>33THENX=1
240 P=P+1:IFP=1THENP=-1:GOTO210
250 L=L+1:GOTO210
260 PRINT@964,"ENTER SELECTION"U$;:INPUTB$
270 IFVAL(B$)=0THEN260ELSEIFVAL(B$)=CTHEN40ELSEIFVAL(B$
     )=C+1THENRUN"MENUMTC/BAS"
275 IFVAL(B$)=C+2THEN400
280 S1$=FS$(VAL(B$))+":"+D$:CLS:PRINT@475,"LOADING "F$(
     VAL(B$))
282 IF LEFT$(S1$,1)<>"&"THEN290
283 S1$=RIGHT$(S1$,(LEN(S1$)-1))
284 CMD "I",S1$
290 RUN S1$
300 PRINT@(L*64+X+32),C"  ""ANOTHER DISK MENU"
310 PRINT@(L*64+X+64),C+1"  ""MENU MAINTENANCE"
315 PRINT@(L*64+X+96),C+2" ""PRINTED MENU"
320 RETURN
400 CLS:PRINT"TURN PRINTER ON AND POSITION PAPER":GOSUB
     15:LPRINT
410 INPUT"ENTER NAME OR NUMBER OF YOUR DISK";H$
411 PRINT@464,"NOW PRINTING"
420 LPRINTTAB(10)"MENU FOR "H$:LPRINT:LPRINT:Y=0
430 FORI=1TOC-1
440 LPRINTTAB(Y)ITAB(Y+4)F$(I);
450 Y=Y+32:IFY>32THENY=0:LPRINT:LPRINT
460 NEXTI:LPRINT:I=0:CLS:PRINT"FINISHED":GOSUB30:CLS:GO
     TO200
```

Program Listing 2

```
1 REM     * COPYRIGHT DR. LARRY HEWIN
2 REM     * VOLKSMICRO COMPUTER SYSTEMS INC.
3 REM     * 202 PACKETS COURT
4 REM     * WILLIAMSBURG, VA 32185
5 REM     * DECEMBER 15, 1979
6 REM     ************************************
7 REM     * DISK MENU SYSTEM - PART # 2      *
8 REM     * MENU MAINTENANCE (FS=MENUMTC/EAS) *
9 REM     ************************************
10 CLEAR 5000:CLS:DEFINT C,I,L,N,P,S,X:DIM F$(26),FS$(2
    6):U$="==>":U1$=CHR$(30)
11 GOTO20
15 FORI=1TO1000:NEXT:RETURN
17 IFN=0THENGOSUB300ELSEGOSUB400
18 RETURN
20 PRINT@471,"DISK MENU SYSTEM":PRINT@597,"BY DR. LARRY
     M. HEWIN"
30 GOSUB15:CLS:PRINT@210,"MENU MAINTENANCE":PRINT@325,"
    ENTER DRIVE ON WHICH MENU DATA IS TO BE PLACED"U$;
    :INPUT D$
35 GOTO600
40 CLS:PRINT@389,"NOW PREPARE TO INPUT A FUNCTIONAL OR
    OTHER TITLE OF":PRINT@453,"THE PROGRAMS YOU WANT T
    O CALL, AND THE FILESPEC.":PRINT@517,"WHEN THROUGH
    , ENTER 'END' FOR FUNCTION OR TITLE":GOSUB15:CLS
50 C=C+1:PRINT@90,"MENU ENTRIES":PRINT@133,"FOR PROGRAM
     #";C;":"
60 PRINT@260,"ENTER PROGRAM TITLE";UI$;U$;:INPUT F$(C)
65 IFF$(C)="END"THEN130
70 PRINT@388,"ENTER FILESPEC      ";U1$;U$;:INPUT FS$(C)
80 PRINT@260,"THE PROGRAM TITLE IS:   "F$(C)
90 PRINT@388,"THE FILESPEC IS:        "FS$(C)
100 PRINT@516,"ARE THESE CORRECT - Y/N ";U$;:INPUTR$
110 IFR$="N"THEN60ELSEIFR$><"Y"THENPRINT"PLS REDO":GOTO
     100
120 CLS:GOTO50
130 R$="":N=C-1:CLS:PRINT@270,"YOU HAVE "N" PROGRAMS FO
     R THE MENU"
140 PRINT@398,"ARE YOU READY TO SAVE TO  DISK - Y/N "U$
     ;:INPUT R$
145 IFR$="N"THEN600ELSEIFR$<>"Y"THEN140
150 PRINT@526,"DATA WILL BE SAVED ON DISK ";D$;" Y/N ";
     U$;:INPUT R$
160 IF R$="N"THENPRINT@654 "";:INPUT"WHAT DISK NO.";D$:
     GOTO170
165 IF R$<>"Y"THEN150
170 S$="MENU/DAT:"+D$:OPEN "O",I,S$
175 CLS:PRINT@465,"SAVING FILE TO DISK";" NO. "D$
180 PRINT#1,N:FORI=1TON:PRINT#1,F$(I);",";FS$(I):NEXTI:
     CLOSE
190 CLS:PRINT@477,"DONE":GOSUB15:RETURN
300 S$="MENU/DAT:"+D$:OPEN "I",1,S$
310 CLS:PRINT@464,"LOADING DATA FROM DRIVE "D$:INPUT#1,
     N
320 FORI=1TON:INPUT#1,F$(I):LINEINPUT#1,FS$(I):PRINT@60
     0,"RECORD # "I;:NEXTI:CLOSE:CLS:PRINT@477,"DONE":G
     OSUB15:CLS
400 PRINT@30,"MENU":L=2:C=1:X=1:P=0
410 PRINT@(L*64+X),C"  "F$(C)
420 C=C+1:IFC>NTHEN445
425 X=X+32:IFX>33THENX=1
430 P=P+1:IFP=1THENP=-1:GOTO410
440 L=L+1:GOTO410
445 C=C-1:RETURN
600 CLS:PRINT@94,"MENU MAINTENANCE FUNCTIONS":PRINT@198
     ,"NEW MENU",,1:PRINT@326,"EDIT MENU",,2:PRINT@454,
     "DELETIONS",,3:PRINT@582,"ADD PROGRAMS",4:PRINT@71
     0,"RUN MENUMGR",5
610 PRINT@862,"ENTER SELECTION"U$;:INPUTS
615 IF S=1THENC=0
620 IFS>5THEN610ELSEIFS<0THEN610
630 ON S GOSUB40,690,790,900,1000
640 GOTO600
690 CLS:GOSUB17
700 PRINT@962,"ENTER ITEM TO BE EDITED - IF NO MORE, EN
     TER ANY LETTER"U$;:INPUTB$:I=VAL(B$):IFI=0THENCLS:
     GOSUB140:RETURN
710 CLS:GOSUB750
715 PRINT@548,"ENTER REVISIONS:":PRINT@712,"PROGRAM:  "
     U$;:INPUTF$(I):PRINT@840,"FILESPEC: "U$;:INPUTFS$(

720 CLS:GOSUB750:PRINT@874,"OK -Y/N"U$;:INPUTR$:IFR$="N
     "THEN710ELSECLS:GOSUB400:GOTO700
750 CLS:PRINT@70,"SELECTED ITEM IS NOW:":PRINT@200,"PRO
     GRAM:  "F$(I):PRINT@328,"FILESPEC: "FS$(I):RETURN
790 CLS:GOSUB17
800 PRINT@960,"ENTER ITEM TO BE DELETED - IF NO MORE, E
     NTER ANY LETTER"U$;:INPUTB$:I=VAL(B$):IFI=0THENCLS
     :GOSUB140:RETURN
810 CLS:GOSUB750:PRINT@618,"IS THIS THE ITEM TO BE DELE
     TED - YES/NO"U$;:INPUTB$
820 IFB$="NO"THENCLS:GOSUB400:GOTO800
825 IFB$<>"YESTHEN810
830 FORC=ITON:F$(C)=F$(C+1):FS$(C)=FS$(C+1):NEXTC:C=0:N
     =N-1:CLS:GOSUB400:GOTO800
900 CLS:GOSUB17:C=N:GOSUB1020:GOSUB40:CLS:RETURN
1000 S$="MENUMGR/BAS"
1010 RUNS$
1020 INPUT"PRESS ENTER TO CONTINUE";Z$:RETURN
```

"DOS to BASIC"). I expect that many readers may already have used it, since it's short and easy. For the convenience of those who haven't, a digest of the instructions, modified to the present purpose, is reproduced in Example 1.

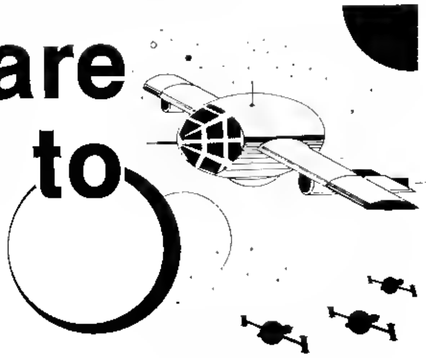
This provides a command program called MENU/CMD. It can be automated by typing AUTO MENU and hitting Enter while in the DOS mode. Hitting reset will then automatically execute MENU/CMD. This will then load BASIC, and run the BASIC program named MENUMGR/BAS. Thus, anytime we hit the reset, we will go back to the disk menu.

This system has solved one of my problems and I've been delighted with the ability to quickly throw a couple of diskettes in the drives, hit reset, and select from 20 to 30 programs. It may be just as useful to other TRS-80 users who have six to twelve diskettes full of programs used frequently.

The program is available from the authors on tape or diskette. ■

✓See List of Advertisers on page 306

TRS-80* Software From Function to Fantasy



POPULAR GAMES

BEGINNER'S BACKGAMMON/KENO—Why sit alone when you can play these fascinating games: ● BACKGAMMON: Play against the computer in a game that's sure to sharpen your skills; ● KENO: Enjoy this popular Las Vegas gambling game—guess the right numbers and win big! **(T1) Order No. 0004R $7.95. Model III compatible**

CHESSMATE-80—This versatile chess opponent gives you a choice of ten levels of play, from the "blitz" level (the computer has 3 seconds to move) to the infinity level (where the computer will consider every possible move—which could take years). This machine-language program is a conservative player and follows all the rules of international play. CHESSMATE-80 can teach you how to move and allow you to set up the board and play end games or special problems. CHESSMATE-80 battled Sargon II to a draw at two minutes a move and beat Microchess 1.5 in six moves. **(T1) Order No. 0057R $19.95. Model III compatible**

YOUR CRIBBAGE AND CHECKERS PARTNER—CRIBBAGE is a two-person game that you are sure to enjoy. This is NOT a tutorial—it is a game worthy adversary. CHECKERS: An old favorite which follows international rules, including multiple jumps. **(T1) Order No. 0068R $9.95. Model III compatible**

CARDS—A one-player package to let you play, with your computer, these famous games: ● DRAW AND STUD POKER: These programs will keep your game sharp; ● NO-TRUMP BRIDGE: Develop your strategy and (hopefully) increase your skill. **(T1) Order No. 0063R $7.95. Model III compatible**

FLIGHT SIMULATIONS

RAMROM PATROL/TIE FIGHTER/KLINGON CAPTURE—•RAMROM PATROL: Destroy the RamRom ships before they capture you. •TIE FIGHTER: Wipe out the enemy Tie fighters and become a hero of the Rebellion. •KLINGON CAPTURE: You must capture the Klingon ship intact. **(T1) Order No. 0028R $7.95. Model III compatible**

FLIGHT PATH—This three-part package includes: •MOUNTAIN PILOT: Become a daring bush pilot and fly supplies to a remote mining camp. You must cross mountain ranges and struggle with headwinds, tricky navigation and rapidly diminishing fuel. •O'HARE: A control tower simulation for you would-be Air Traffice Controllers. You are responsible for the lives of hundreds of passengers as you guide aircraft through your control sector. •PRECISION APPROACH RADAR: Combines the skills of pilot and Air Traffic Controller, as your commands guide an aircraft in its approach to the field and a safe landing. **(T1) Order No. 0171R $9.95. Model III compatible**

BALL TURRET GUNNER—Imagine yourself at the control console of a strategic laser weapon, deep in the space lanes. Your hindsight detector informs you of a Gnat fighter coming in for an attack so you swivel you laser turret until you can see the target. Watch the Range Indicator and your Targeting Computer's readout closely, because you'll only have a fraction of a second to catch him in your sights. Will you transform the Gnat into a ball of ionized gas or will you see that blinding flash that means The Big Demotion? BALL TURRET GUNNER, with you choice of multiple levels of difficulty, optional sound effects and excellent graphics, is more than a game. It's an event to be savored. **(T1) Order No. 0051R $9.95.**

JET FIGHTER PILOT—In this brilliantly realistic simulation, you become the pilot of a twin turbo-jet fighter. Begin your mission from either the deck of a carrier or from an airfield. During flight, you'll need to constantly monitor your display and make the necessary adjustments to the throttle, flaps, and air spoilers; you must decide when to retract landing gear and release your drop tanks! There is an on-board Navigational Computer, a Glideslope/Localizer and a Weapons Control Computer. Earn your wings with JET FIGHTER PILOT. **(T1) Order No. 0159R $14.95. Model III compatible**

SPACE TREK II—Protect the quadrant from the invading Klingon warships. The Enterprise is equipped with phasers, photon torpedoes, impulse power and warp drive. **(T1) Order No. 0002R $7.95. Model III compatible**

AIR FLIGHT SIMULATION—Take off and land your aircraft without making a crater. This "instruments only" simulation starts you with a full tank of fuel, which gives you a maximum range of about 50 miles. You'll get constant updates of air speed, compass heading and altitude. After you've acquired a few hours of flight time, you can try flying a course against a map or doing aerobatic maneuvers. **T(1) Order No. 0017R. $9.95. Model III compatible**

SPACE TREK IV—•STELLAR WARS: Engage and destroy Tie fighters in your attack on the Death Star. For one player. •POPULATION SIMULATION: A two-player game where you control the economy of two neighboring planets. You must decide: Guns or Butter? **(T1) Order No. 0034R $7.95. Model III compatible**

BASIC AND INTERMEDIATE LUNAR LANDER—Bring your lander in under manual control. The basic version is for beginners; the intermediate version is more difficult, with a choice of landing areas and rugged terrain. **(T1) Order No. 0001R $7.95. Model III compatible**

COSMIC PATROL—We put you in command of a small interstellar patrol craft. You must defend Terran space and prey on the Quelon freighters that carry vital war supplies—but beware of their I-Fighter escorts. They're well armed, extremely fast and they NEVER miss! With its real-time action, impressive sound option and superb graphics, this machine-language program is the best of the genre. **(T1) Order No. 0223R $14.95. Model III compatible**

Airmail Pilot—Return to the early days of aviation. You must fly the mail from Columbus to Chicago. Your Jenny, a cloth-covered biplane, must take you through unpredictable winds, hail and electrical storms. Your mission is to get the mail through in the shortest possible time. There is an on-board clock to time you flight, from takeoff to touchdown...assuming you are able to complete it. **(T1) Order No. 0106R $9.95. Model III compatible**

NIGHT FLIGHT—Your mission is to fly over the North Atlantic and make a nighttime photo/recon flight above the enemy fleet. NIGHT FLIGHT lets you take-off, fly and land a propellar-driven aircraft. You can practice approaches and landings with an on-screen display of the landing field information—it will practically teach you to fly. **(T1) Order No. 0117R $9.95. Model III compatible**

COMP-U-NOVELS

WHO-DUN-IT? Criminal elements have committed five dastardly crimes. As the investigating detective, you must solve them.

You can compete against either Detective Nybbles, a computerized sleuth, or up to four other human detectives.

•DEDUCTION: Guess the order of four symbols out of six or seven different ones. To make things even more complicated, you can let the computer repeat symbols and have a range of 2401 possibilities. **(T1) Order No. 0047R $7.95. Model III compatible**

SANTA PARAVIA AND FIUMACCIO Become the ruler of a medieval city-state as you struggle to create a kingdom. Up to six players can compete to see who will become the King or Queen first. **(T1) Order No. 0043R $7.95. Model III compatible**

There are over 450 Instant Software dealers throughout the U.S.A. and the world.

We usually ship our orders within 48 hrs.

CODE—Minimum System Required

(T1) = TRS-80 Model I Level II, 16K RAM
(T2) = TRS-80 Model I Level II, 16K RAM with Expansion Interface 16 + K RAM and one disk drive
(T3) = TRS-80 Model II, 32K RAM

*TRS-80 is a trademark of Tandy Corporation.

SEE YOUR LOCAL INSTANT SOFTWARE DEALER OR

Just Call Toll-Free 1-800-258-5473 Orders Only

IN NEW HAMPSHIRE DIAL 1-603-924-7296

Write for Our New Instant Software Catalog

Instant Software™

PETERBOROUGH, N.H. 03458

✓See List of Advertisers on page 306

A seagoing simulation of the action in the Bungoe Straits.

Subdestroy

John Cominio
626 Tortoise Way
Satellite Beach, FL 32937

When I go to a shopping mall the first place I head for is the games arcade. One day I discovered a new game that involved dropping depth charges from a destroyer onto submarines. I was fascinated with the game and wanted to play it longer but I ran out of change.

Since I enjoyed the game so much I decided to try to program my TRS-80 to simulate it. A couple of days later I emerged from my room after completing the programming for my version of the game, which I named Sub-Destroy.

The Program

Sub-Destroy is written for a 16 K Level II TRS-80. It doesn't contain any special machine language subroutines so answer the Memory Size? prompt with Enter.

After you have loaded the program and typed Run you will be asked if you want instructions. They will remain on the screen until you press Enter. After pressing Enter, the screen will clear and you will see your destroyer cruising on the ocean's waterline with three submarines traveling underneath.

The object of the game is to score as many possible points as you can with only twenty depth charges. If you score well enough, you will receive bonus depth charges at the end of the game.

There are three different types of submarines; each varies in point value. The largest and fastest is worth thirty points, the medium size sub is worth ten points, and the smallest is worth five points.

When you drop a depth charge, you must set the level at which it is to explode with the up or down arrow. A line will then flash across the screen to indicate the depth at which the charge will explode.

When a depth charge is dropped all the ships will move faster and you will have to compensate for this. In addition, when a depth charge is dropped it will move forward in order to try and keep pace with the destroyer.

If you hit a ship, it will sink and your score and charges left indicators will be updated.

At the end of the game, if you score above 35 points you will receive extra depth charges. Depending on your score, you will receive up to twenty extra charges or one replay.

I haven't found any problems in the program. If you do have problems after typing it in, I suggest you check the number of spaces in the strings, for example, lines 350,400,530–550, etc.■

B$, B1$, B2$-	submarine Chr$'s
E$-	depth indicator Chr$
E1$, E2$-	erase strings
J-	score
J1-	number of depth charges
O-	destroyers print@ position
X-	explode depth
Z, S, A-	submarine print@ positions
S$-	destroyers Chr$

Fig. 1. Variables

Program Listing

```
10 CLEAR400
20 DEFINTA-Z
30 RANDOM
40 ONERRORGOTO720
50 CLS
60 PRINTTAB(15)"--------- S U B  D E S T R O Y ---------":PRINT:PR
INT:PRINT"DO YOU WANT INSTRUCTION (Y/N) ?"
70 R$=INKEY$:IFR$=""THEN70
80 IFR$="Y"THEN1220ELSEIFR$="N"THENCLS:GOTO90ELSE70
90 Q=1:J1=20
100 GOSUB110:GOTO250
110 E1$=STRING$(4,128)
120 E2$=STRING$(6,128)
130 A=RND(190):S=A+RND(150)
140 Z=RND(1024):IFZ>1024ORZ<384THEN130
150 IFZ+A>1024ORZ+A<384THEN130
160 IFZ+A+S>1024ORZ+A+S<384THEN130
170 X=512
180 L$=STRING$(63,140)
190 B2$=CHR$(176)+CHR$(176)+CHR$(184)+CHR$(181)+CHR$(176)+CHR$(1
76)
200 E$=STRING$(63,128)
210 B$=CHR$(176)+CHR$(187)+CHR$(183)+CHR$(176)
220 B1$=CHR$(176)+CHR$(176)+CHR$(183)+CHR$(176)
230 S$=CHR$(176)+CHR$(184)+CHR$(176)+CHR$(186)+CHR$(176)+CHR$(18
4)+CHR$(176)
240 RETURN
250 GOSUB260:GOTO270
260 PRINT@64,STRING$(64,45);:RETURN
270 A$=INKEY$:IFA$=""THENGOSUB260:GOTO280
280 GOSUB750
290 IFA$=CHR$(32)THENQ1=0:J1=J1-1:GOSUB400
300 GOSUB350
310 GOTO270
320 END
330 GOSUB750
340 RETURN
350 IFQ>57THENQ=1:PRINT@57,"        ";
360 GOSUB700
```

Program continues


How's your love life?
INTERLUDE
THE ULTIMATE
EXPERIENCE...
A little dull around the edges? Routine? Predictable? Boring? Maybe all it needs is a little Interlude. Interlude is the most stimulating computer game ever conceived. It combines a computer interview, an innovative programming concept, and a one-of-a-kind manual to turn your love life into exciting, adventurous, delicious fun!
Interlude is: romantic... playful... outrageous... a fantasy. Interlude is: ■ A Bed of Roses (Interlude #1) ■ Mata Hari (Interlude #49) ■ The Chase (Interlude #7) ■ Rodeo! (Interlude #71) ■ The King and I (Interlude #60) ■ Some Enchanted Evening (Interlude #84) ■ Caveman Caper (Interlude #82) ■ From Here to Ecstasy (Interlude No. 30) ■ Satin Dreams (Interlude #72).
More than 100 Interludes are included in the program. Most are described in detail in the accompanying manual, but several surprise Interludes are buried in the program awaiting that very special time when your interview says you're ready. (When you learn secret Interlude #99, your love life may never again be the same!) Interlude can give you experiences you'll never forget. Are you ready for it?
Interlude™
The Ultimate Experience.
287
INTERLUDE, 10635 Richmond, Houston, Texas 77042. I'm really ready. Send my Interlude today.
Apple II (16K)*
☐ Cassette ($16.95)
☐ Diskette ($19.95)
☐ Diskette—Pascal or DOS 3.3 ($19.95)
Add $1.50 for shipping and handling.
TRS-80 (Level II-16K)**
☐ Cassette ($16.95)
☐ Diskette ($19.95)
Poster
☐ 20"x 24" reproduction of this ad without ad copy ($4.95—includes shipping charges)
Available for immediate shipment.
Please enclose your check payable to INTERLUDE or complete the charge information:
☐ MASTERCARD ☐ VISA All charge customers must sign here ____
Account No. ____ Expiration date ____ MasterCard Bank Code ____
CHARGE CUSTOMERS: Order by phone toll-free! **1-800-231-5768 Ext. 306** (Texas: 1-800-392-2348 Ext. 306)
Name ____ Age ____
Address ____
City ____ State ____ Zip ____
*Apple II is a registered trademark of Apple Computers, Inc. **TRS-80 is a registered trademark of Radio Shack, a Tandy Co.

```
370 GOSUB440
380 PRINT@Q-1,E1$::PRINT@Q,S$;:Q=Q+1:RETURN
390 GOTO700
400 IFQ+Q1+64>XTHENPRINT@Q+Q1+64,"****"::FORR=1TO10:NEXTR:PRINT@
Q+Q1+64,"    ";:RETURNELSE410
410 GOSUB620:PRINT@Q+Q1+64,CHR$(191);
420 GOSUB440
430 PRINT@Q+Q1+64,CHR$(128);:Q1=Q1+65:GOTO400
440 PRINT@Z+4,B$;:Z=Z+1:GOSUB330
450 PRINT@Z-1,E1$;
460 IFZ+A+S>1005THEN520
470 PRINT@Z+S+A+4,B2$;:S=S+1:GOSUB330
480 PRINT@Z+S+A-1,E1$;
490 PRINT@Z+A+4,B1$;:A=A+1:GOSUB330
500 PRINT@Z+A-1,E1$;
510 RETURN
520 PRINT@960,STRING$(63,128);
530 PRINT@Z+3,"    ";
540 PRINT@Z+A+2,"    ":
550 PRINT@Q+Q1+64," ";
560 GOSUB620
570 Z=0:A=0:S=0:Z=RND(1024):IFZ>1024ORZ<384THEN570
580 A=RND(190):S=RND(150)
590 IFZ+A>1024ORZ+A<384THEN570
600 IFZ+A+S>1024ORZ+A+S<384THEN580
610 GOTO270
620 GOTO630
630 Z2=Z+A+S+4:Z1=Z+A+4:Z4=Z+4
640 Q3=Q+Q1+64:Q3=Q3+64
650 IFQ3>=XANDQ3<=X+64THEN660ELSERETURN
660 IFQ3>=Z4-4ANDQ3<=Z4+4THENJ=J+5:GOSUB950:GOTO690ELSE670
670 IFQ3>=Z1-4ANDQ3<=Z1+4THENJ=J+10:GOSUB1040:GOTO690ELSE680
680 IFQ3>=Z2-6ANDQ3<=Z2+6THENJ=J+30:GOSUB1130:GOTO690ELSERETURN
690 Q1=0:Z2=0:Z1=0:Z4=0:A$="":GOSUB700:FORT=1TO1000:NEXT:GOTO280
700 PRINT@128,"SCORE=";J;:PRINT@142,"CHARGES LEFT=";J1;
710 IFJ1=0THEN730ELSERETURN
720 Q1=0:RESUME270
730 GOTO830
740 GOTO740
750 IFA$="["THEN760ELSE790
760 IFX=320THENX=X+64
770 X=X-64:PRINT@X,L$;:FORR=1TO5:NEXTR:PRINT@X,E$;
780 A$="":RETURN
790 IFA$=CHR$(10)THEN800ELSERETURN
800 IFX=960THENX=X-64
810 X=X+64:PRINT@X,L$;:FORR=1TO5:NEXTR:PRINT@X,E$;
820 A$="":RETURN
830 CLS
840 IFJ<35THENPRINT@0,"THE GAME IS OVER.  YOU HAD";J+JA;"TOTAL P
OINTS.";:GOTO840ELSE850
850 IFJ>34ANDJ<=150THENJ1=2
860 IFJ>150ANDJ<200THENJ1=6
870 IFJ>200ANDJ<=300THENJ1=9
880 IFJ>300ANDJ<=500THENJ1=14
890 IFJ>500THENJ1=20
900 PRINT@0,"THE GAME IS OVER, BUT YOU ARE LUCKY."
910 PRINT"YOUR SCORE WAS";J;"AND THAT ENTITLES YOU TO";J1;"BONUS
 ";
920 PRINT"MISSLES."
930 PRINT"CREDIT GAME WILL RESUME WHEN THE TIMER REACHES ZERO."
940 FORR=1000TO0STEP-1:PRINT@512,R;:NEXT:CLS:Q=1:JA=J:J=0:GOTO11
0
950 FORP=1TO10:PRINT@Z4-64,"GLUG";
960 IFZ4>=1015THENPRINT@Z4-64,"    ";:RETURN
970 PRINT@Z4,B$;
980 FORR=1TO100:NEXT
990 PRINT@Z4-64,"    ";
1000 PRINT@Z4-1,E1$+" ";
1010 FORR=1TO100:NEXT
1020 Z4=Z4+64
1030 NEXTP
1040 FORP=1TO10:PRINT@Z1-64,"GLUG";
1050 IFZ1>=1015THENPRINT@Z1-64,"    ";:RETURN
1060 PRINT@Z1,B1$;
1070 FORR=1TO100:NEXTR
1080 PRINT@Z1-64,"    ";
1090 PRINT@Z1-1,E1$+" ";
1100 FORR=1TO100:NEXTR
1110 Z1=Z1+64
1120 NEXTP
1130 FORP=1TO10:PRINT@Z2-64,"GLUG";
1140 IFZ2>=1015THENPRINT@Z2-64,"    ";:RETURN
1150 PRINT@Z2,B2$;
1160 FORR=1TO100:NEXTR
1170 PRINT@Z2-64,"    ";
1180 PRINT@Z2-2,E2$+"  "::
1190 FORR=1TO100:NEXTR
1200 Z2=Z2+64
1210 NEXTP
1220 CLS:PRINTTAB(10)"*********** S U B  D E S T R O Y *********
**"
1230 PRINT:PRINTTAB(5)"THE OBJECT OF THIS GAME IS TO TRY AND SIN
K THE SUBMARINES"
1240 PRINT"TRAVELING BELOW YOU.  YOU CAN ACOMPLISH THIS BY DROPP
ING DEPTH"
1250 PRINT"CHARGES FROM YOUR DESTROYER SHIP CRUISING AT THE SURF
ACE OF THE"
1260 PRINT"OCEAN.  YOU MUST SET THE LEVEL WHERE YOU WANT THE DEP
TH CHARGE"
1270 PRINT"TO EXPLODE.  TO DO THIS YOU POSITION A WHITE LINE AT
THE PROPER"
1280 PRINT"DEPTH.  TO MOVE THE LINE YOU MAY USE THE UP ARROW ([)
 OR THE"
1290 PRINT"DOWN ARROW (";CHR$(92);").";
1300 PRINT" TO DROP YOUR CHARGES JUST PRESS THE SPACE BAR."
1310 PRINTTAB(5)"SCORING IS AS FOLLOWS:"
1320 PRINTTAB(10)"30 POINTS FOR THE LARGE SUBMARINE"
1330 PRINTTAB(10)"10 POINTS FOR THE SMALLER SUBMARINE"
1340 PRINTTAB(10)"5  POINTS FOR THE SLOWEST SUBMARINE"
1350 PRINTTAB(5)"IF YOU SCORE WELL ENOUGH YOU WILL RECEIVE BONUS
 CHARGES."
1360 PRINT"THE HIGHER YOU SCORE THE MORE BONUS CHARGES YOU WILL
RECEIVE.   REMEMBER YOU ONLY HAVE 20 CHARGES TO START WITH.  PRE
SS ENTER.";
1370 R$=INKEY$:IFR$=""THEN1370
1380 IFR$=CHR$(13)THENCLS:GOTO90:60ELSE1370
1390 END
```

★★

4MHZ, DOUBLE DENSITY, COLOR&B/W GRAPHICS . . THE LNW80 COMPUTER

COMPARE THE FEATURES AND PERFORMANCE

FEATURES	LNW80	PMC-80**	TRS-80* MODEL III
PROCESSOR	4.0 MHZ	1.8 MHZ	2.0 MHZ
LEVEL II BASIC INTERP.	YES	YES	LEVEL III BASIC
TRS80 MODEL I LEVEL II COMPATIBLE	YES	YES	NO
48K BYTES RAM	YES	YES	YES
CASSETTE BAUD RATE	500/1000	500	500/1500
FLOPPY DISK CONTROLLER	SINGLE/ DOUBLE	SINGLE	SINGLE/ DOUBLE
SERIAL RS232 PORT	YES	YES	YES
PRINTER PORT	YES	YES	YES
REAL TIME CLOCK	YES	YES	YES
24 X 80 CHARACTERS	YES	NO	NO
VIDEO MONITOR	YES	YES	YES
UPPER AND LOWER CASE	YES	OPTIONAL	YES
REVERSE VIDEO	YES	NO	NO
KEYBOARD	63 KEY	53 KEY	53 KEY
NUMERIC KEY PAD	YES	NO	YES
B/W GRAPHICS, 128 X 48	YES	YES	YES
HI-RESOLUTION B/W GRAPHICS, 480 X 192	YES	NO	NO
HI-RESOLUTION COLOR GRAPHICS (NTSC), 128 X 192 IN 8 COLORS	YES	NO	NO
HI-RESOLUTION COLOR GRAPHICS (RGB), 384 X 192 IN 8 COLORS	OPTIONAL	NO	NO
WARRANTY	6 MONTHS	90 DAYS	90 DAYS
TOTAL SYSTEM PRICE	$1,664.00	$1,840.00	$2,187.00
LESS MONITOR AND DISK DRIVE	$1,200.00	$1,375.00	---

When you've compared the features of an LNW80 Computer, you'll quickly understand why the LNW80 is the ultimate TRS80 software compatible system. LNW RESEARCH offers the most complete microcomputer system at an outstanding low price.
We back up our product with an unconventional 6 month warranty and a 10 days full refund policy, less shipping charges.

LNW80 Computer $1,200.00
LNW80 Computer w/B&W Monitor & one 5" Drive $1,664.00
All orders must be prepaid, CA residents please include 6% sales tax.
Shipping and handling charge of $15.00 must be included with every order.
* TRS80 Product of Tandy Corporation.
** PMC Product of Personal Microcomputer, Inc.

LNW80

- BARE PRINTED CIRCUIT BOARD & MANUAL $89.95

The LNW80 - A high-speed color computer totally compatible with the TRS-80*. The LNW80 gives you the edge in satisfying your computation needs in business, scientific and personal computation. With performance of 4 MHz, Z80A CPU, you'll achieve performance of over twice the processing speed of a TRS-80*. This means you'll get the performance that is comparable to the most expensive microcomputer with the compatibility to the world's most popular computer (TRS-80*) resulting in the widest software base.

FEATURES:
- TRS-80 Model I Level II Software Compatible
- High Resolution Graphics
  - RGB Output - 384 x 192 in 8 Colors
  - NTSC Video or RF MOD - 128 x 192 in 8 Colors
  - Black and White - 480 x 192
- 4 MHz CPU
- 500/1000 Baud Cassette
- Upper and Lower Case
- 16K Bytes RAM, 12K Bytes ROM
- Solder Masked and Silkscreened

LNW SYSTEM EXPANSION

- BARE PRINTED CIRCUIT BOARD AND MANUAL $69.95
WITH GOLD CONNECTORS $84.95

The System Expansion will allow you to expand your LNW80, TRS-80*, or PMC-80** to a complete computer system that is still totally software compatible with the TRS-80* Model I Level II.

FEATURES:
- 32K Bytes Memory
- 5" Floppy Controller
- Serial RS232 20ma I/O
- Parallel Printer
- Real Time Clock
- Screen Printer Bus
- On Board Power Supply
- Solder Masked and Silkscreened

KEYBOARD

LNW80 KEYBOARD KIT $84.95

The Keyboard Kit contains a 63 key plus a 10 key, P.C. board, and remaining components.

LNDoubler + DOS Plus 3.2D

- Assembled and Tested $175.00

Double-density disk storage for the LNW Research's "System Expansion" or the Tandy's "Expansion Interface". The LNDoubler™ is totally software compatible with any double density software generated for the Percom's Doubler***. The LNDoubler™ provides the following outstanding features.

- Store up to 350K bytes on a single 5" disk
- Single and double density data separation
- Precision write precompensation circuit
- Software switch between single and double density
- Hardware override into single density only
- Easy plug in installation requiring no etch cuts, jumpers or soldering
- 35, 40, 77, 80 track 5" disk operation
- 120 day parts and labor Warranty

*** Doubler is a product of Percom Data Company, Inc.

DOS PLUS 3.2D

Micro Systems software's double density disk operating system. This operating system contains all the outstanding features of a well developed DOS, with ease in useability.

LNW DATA SEPERATOR

- Assembled and Tested $17.95

The LNW Data Separator provides you with a reliable and inexpensive means of solving your disk data read error problems for your 5" single density drives. Compatible with both the LNW System Expansion and Tandy's Expansion Interface. Some soldering is required.

CASE

LNW80 CASE . Add $12.00 for postage $84.95

The streamline design of this metal case will house the LNW80, LWN System Expansion, LNW80 Keyboard, power supply and fan, LNDoubler™, or LNW Data Separator. This kit includes all the hardware to mount all of the above.

PARTS AVAILABLE FROM LNW RESEARCH
4116 - 200ns RAM
6 chip set $26.00
8 chip set $33.50
16 chip set $64.00
24 chip set $94.00
32 chip set $124.00

- LNW80 "Start up parts set" LNW80-1 . . . . $82.00
- LNW80 "Video parts set" LNW80-2 . . . . $31.00
- LNW80 Transformer LNW80-3 . . . . $18.00
- LNW80 Keyboard cable LNW80-4 . . . . $16.00
- 40 Pin computer to expansion cable $15.00
- System Expansion Transformer $19.00
- Floppy Controller (FD1771) and UART (TR1602) . . . $30.00

LNW RESEARCH ✓53
C O R P O R A T I O N
14661-C MYFORD RD.
TUSTIN CA. 92680

VISA & MASTER CHARGE ACCEPTED, Add $3 for postage
ORDERS & INFO. NO. 714-552-8946
SERVICE NO. 714-641-885D

✓See List of Advertisers on page 306

ENCYCLOPEDIA FOR THE TRS-80*
ENCYCLOPEDIA FOR THE TRS-80*
THEORY
PRACTICE
PROGRAMS
VOLUME 1
*Trademark of Tandy Corp

This routine will convert your uppercase text files to upper/lowercase. Period.

CAPTRAN

Buzz Gorsky K8BG
712 Hillside Drive
Carlisle PA 17013

At one time or another every computer owner thinks about a word processing program for his or her machine. And often, as your hardware system grows,. the uses for word processing also grow. At least that was true for me.

I began using a word processor with my 16K system and an ancient (but very functional) Model 15 teletype machine. At that time there was little reason to worry about upper and lowercase since the printer had only uppercase characters. As I've become more experienced with word processing, I hope to obtain an ASCII printer with the full character set. Consequently, I am reworking the system to handle upper and lowercase.

This article describes a program I wrote to convert text files of only uppercase characters to files with upper and lowercase.

Unpredictable Text

It is impossible to write a program that will completely convert a text file, since the presence of all capitals in a text is not predictable. However, 99 percent of the work can be done automatically. This program was written to run with my word processor. It resides in high memory, out of the way of my word processing program.

The program first converts all uppercase ASCII characters to lowercase and then converts the first letter character following a period to an uppercase character. Here's how it works.

The variables BUFFER and BUFSIZ correspond to my word processor program and provide the location and size of the text buffer area in memory. This program begins at FFA0H. Starting in line 180 things are set up to find ASCII uppercase letters; that is, characters with an ASCII value greater than 64. The CPI function takes care of the comparisons. If parity is odd after the CPI, then the entire buffer set has been scanned so we go to PASS 2. If the result of the compare was negative, indicting an ASCII character, go to TRAN; otherwise we loop back for another compare.

In TRAN we DEC HL back to the ASCII character (CPI increments HL after the compare is completed) and SET the fifth bit in the location pointed to by HL. This changes the value to that of a lowercase character. Next, increase HL back to its value on entering TRAN and then loop back to the compare. In this way COMP and TRAN take care of changing capital letters to lowercase letters.

Once this sequence is completed, the program will continue with PASS 2. Here, things are set up to look for the period (ASCII 46). After the CPIR, the ZERO flag will be set if the compare was true, that is if HL points to a period. Otherwise the CPIR would be complete when the entire buffer area had been scanned. In line 350, if the ZERO flag has not been set, we branch to 402D in order to reenter DOS. Otherwise when a period is identified we will be in TRAN1.

There, we put the value now pointed to by HL (this will be the next character after the period)

```
                     00100 ;Z/Z/Z/Z/Z/Z/Z/Z/Z/Z/Z/Z/Z/Z/Z/Z/Z/Z/Z/
                     00110
                     00120 ;      C A P T R A N    B Y   K 8 B G
                     00130
                     00140 ;Z/Z/Z/Z/Z/Z/Z/Z/Z/Z/Z/Z/Z/Z/Z/Z/Z/Z/
A000                 00150 BUFSIZ  EQU     0A000H
5500                 00160 BUFFER  EQU     5500H
FFA0                 00170         ORG     0FFA0H
FFA0 210055          00180 BEG     LD      HL,BUFFER
FFA3 0100A0          00190         LD      BC,BUFSIZ
FFA6 3E40            00200         LD      A,64
FFA8 EDA1            00210 COMP    CPI
FFAA E2B8FF          00220         JP      PO,PASS2
FFAD FAB2FF          00230         JP      M,TRAN
FFB0 18F6            00240         JR      COMP
                     00250
FFB2 2B              00260 TRAN    DEC     HL
FFB3 CBEE            00270         SET     5,(HL)
FFB5 23              00280         INC     HL
FFB6 18F0            00290         JR      COMP
                     00300
FFB8 210055          00310 PASS2   LD      HL,BUFFER
FFBB 0100A0          00320         LD      BC,BUFSIZ
FFBE 3E2E            00330 COM     LD      A,46
FFC0 EDB1            00340         CPIR
FFC2 C22D40          00350         JP      NZ,402DH
FFC5 7E              00360 TRAN1   LD      A,(HL)
FFC6 B7              00370         OR      A
FFC7 280C            00380         JR      Z,BAK
FFC9 FE0D            00390         CP      13
FFCB 2808            00400         JR      Z,BAK
FFCD FE20            00410         CP      20H
FFCF 2804            00420         JR      Z,BAK
FFD1 CBAE            00430         RES     5,(HL)
FFD3 18E9            00440         JR      COM
                     00450
FFD5 23              00460 BAK     INC     HL
FFD6 0B              00470         DEC     BC
FFD7 79              00480         LD      A,C
FFD8 B0              00490         OR      B
FFD9 20EA            00500         JR      NZ,TRAN1
FFDB C39201          00510         JP      402Dh
FFA0                 00520         END     BEG
00000 TOTAL ERRORS
```

Program Listing

into the A register. We then check to see if this is a zero. (In my word processor, 0 is used as a blank space which is not to be printed.)

If we find a zero, we branch to BAK. If the character is not a zero, we check to see if it is a 13, which is a carriage return. If it is, we branch to BAK.

If not, we check to see if it is a space (20H) and if it is we go to BAK. If it is none of these, we assume that it is a letter. On occasion this will be a false assumption, but it's usually correct. An alternative is to change line 370 to CP 65 and line 380 to JP M,BAK and delete 390–420. This returns to BAK for any ASCII value below 65 and may work better for common text.

Since the letter after a period often starts a new sentence, RESET the fifth bit to restore the character to uppercase and then loop back to COM to continue the comparisons.

What Follows a Period

BAK is a routine which will help us avoid characters that follow a period, but are not letters. We increase HL, decrease BC and then check to see if BC has hit zero. (No zero flag is set when BC is decremented to zero, so we put C in A and then OR B, which will set the zero flag if both B and C are zero.) If it hasn't hit zero, we loop back to TRAN1 to check the next character. If BC is zero, then the task is done and we go to DOS.

When I use the program, first I enter my word processor which loads the text file into memory. Next, I exit that program and go to DOS to run CAPTRAN. When that has run I am back in DOS and can reenter the word processor. As an alternative, this program could be put into the word processor as a function to be called within the program. It's also possible to add a bit to this program to distinguish other marks of punctuation. It depends on what your data files contain. For me, this simple version does a reasonable job. ■

✓See List of Advertisers on page 306

The page that was left out of the Level II manual.

The Level II Index

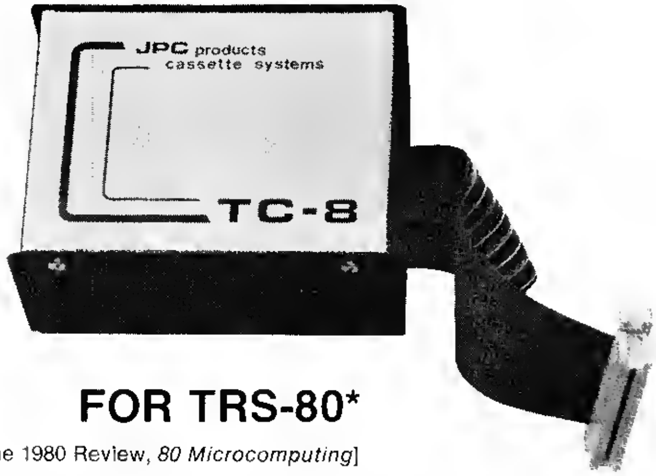
Barbara Mercer, Mercer Systems, Inc., 87 Scooter Lane, Hicksville, NY 11801

Editor's Note: Having spent the last 17 years in data processing where he developed applications on everything from the 1401 to the 3033, Stephen Hughes, President of Mercer Systems, Inc., was embarrassed one day as he flipped through the Level II Basic Reference Manual and found no index!

Barbara Mercer authored the following, supplemental index for their own reference. Feeling that more people could benefit from it, the author decided to share it with *80* readers.

Poor Man's Floppy

HIGH SPEED CASSETTE SYSTEM



FOR TRS-80*

Now the widely acclaimed JPC Cassette System is available for your TRS-80 computer. The price is only $90.00*

[Reprint of June 1980 Review, *80 Microcomputing*]

TC-8 Cassette System
JPC Products
Albuquerque, NM
Kit: $90
Assembled: $120

by Carl A. Kollar

I guess I don't have to tell any TRS-80 owners how frustrating the cassette system that comes with the computer can be. Even with the factory mod that's available, the annoyance of loading and checking programs becomes just barely tolerable.

If you're like me, after you've just plunked down a chunk of money for a Level II 16K machine, "you ain't got nuttin left" for even one disk drive at 500 bucks apiece. So you suffer.

A reasonable alternative is the Exatron Stringy Floppy (ESF). This will cost you about 250 bucks and totally eliminates your loading and saving problems, automatically and fast. I've had one of these for about six months and love it!

But, if the price is still too steep, have I got a device for you!

The Device

The February 1980 issue of *Microcomputing* had an ad that intrigued the hell out of me. It was a high-speed cassette system by JPC Products acclaimed as a "poor man's floppy." It made all sorts of seemingly ridiculous claims such as "loads five times faster," "stores 50,000 bytes on a 10-minute cassette," "less than one bad load in a million bytes with the volume control anywhere between one and eight."

All this for a measly [90] bucks? How could this be? A call to Albuquerque answered a few questions: Yes, it had its own power supply, and, it stored programs five times faster because it utilized higher density data. The computer outputs the information at a higher rate out of the rear keyboard connector.

The ad had even claimed anyone could build it even if you have never soldered before. JPC would make it work, if you couldn't—for free. I was sold. I placed my order, and it arrived about two months later (parts shortage).

I work in electronics, so I found the unit exceptionally easy to build. It took about an hour. The manual is superb. (That's better than great.) It was clear, concise and exact with no ambiguities. Important parts placements are stressed (polarity markings on electrolytics, bands on diodes, etc.).

JPC was right! With these instructions, you couldn't go wrong. The board quality is excellent. It is double-sided and parts locations are clearly marked on the component side of the board. There are no jumper wires to install. JPC utilizes PC traces and plated-through holes for connections to traces on the other side of the board.

Also, there are absolutely no adjustments or settings to bother with.

The documentation is a sheaf of 8½ × 11 papers stapled together. It is written in the nicest format I've seen in a while. Each command and/or subjects is covered on its own sheet in large type. All explanations are in easy to read English—not computerese.

Commands and Features

SAVE"filename": Saves your BASIC program on cassette.
LOAD: Reads the next BASIC program from the cassette.
LOAD"filename": Searches for and loads the specified file from cassette.
LOAD? and LOAD?"filename": Reads file from cassette, and compares contents to memory.
LOADN: Prints a list of all the programs on a cassette, until interrupted by the "break" key.
LOADN"filename": Same as above except the tape will stop at the end of the program named.
KILL: Removes the file manager program from memory so that the extra memory can be used by large programs.
RSET: Allows the operator to rewind and position the tape on tape recorders that have these functions tied to the motor control jack.
RUN"filename": TC-8 searches for a specified program and runs it immediately.
PUT"filename": Same as SAVE "filename", except it is for use with system tapes.
GET: Same as LOAD, except it is for use with system tapes.
GET"filename": Same as LOAD "filename", except it is for use with system tapes.
GET? and GET?"filename": Same as LOAD? and LOAD?"filename", except it is for use with system tapes.
GETN and GETN"filename": Same as LOADN and LOADN"filename", except it is for use with system tapes.
OPEN: Required before cassette input or output of a data file can be attempted.
CLOSE: Required to end a cassette data file.
PRINT#: Allows numerical or string data to be output to a cassette file.
INPUT#: Allows numerical or string data to be input from a cassette file.

I haven't counted them, so I don't know about the "one load in a million bytes" claim, but my son, Anthony (age 11), loaded about 30 of his programs from his Radio Shack format tape to a new TC-8 format tape. He's run them all and found no bad loads.

Unlike the standard tape system, you can position your tape anywhere before the program you want and not have to look for a blank spot between programs. The TC-8 patiently waits for the program you want and then starts loading without getting confused by the portion of the previous program you just fed it.

Try that on your regular cassette system; you'll wear out the reset button. ■

ORDER NOW

To order your TC-8 kit, send your check or money order for $90.00 plus $3.50 postage and handling to JPC PRODUCTS CO., 12021 Paisano Ct., Albuquerque, NM 87112 (New Mexico residents add 4% sales tax). Credit card orders accepted by phone or mail. Personal checks will delay shipment. We will otherwise immediately ship you the TC-8 kit, the cabinet, the ribbon cable, the power adapter, an instruction manual, and a cassette containing the software.

✓190

JPC PRODUCTS CO.
Phone (505) 294-4623
12021 Paisano Ct.
Albuquerque, N.M. 87112

✓See List of Advertisers on page 306

Here are some answers to frequently asked questions.

Model II Q and A

Tom Yager
PO Box 566, Union Station
Endicott, NY 13760

I often receive letters and calls from distraught Model II owners, most of whom are having some difficulty moving up from the Model I.

Here's some of their more frequent questions, with my solutions:

Q: I miss my Model I graphics. Is there any way I can draw with my new computer?

A: Not as well as on the Model I. The grid on the Model II is only 80 horizontal by 24 vertical, about half the size of its predecessor. The best you can do is use the BASIC statements in Fig. 1 in place of SET and RESET.

Q: My disk isn't operating properly. My computer works with other disks, but there's one or two it just refuses to allow me to use.

A: This could be caused by several things, but there are three which I have found most common. First, the disk area where the information you're trying to access is stored could be flawed. Flaws include disks affected by humidity, heat, cold, static electricity, rough handling, and other environmental hazards. If the problem is a foreign object, such as dust or cigarette ash, a few attempts to use the disk might cause the obstruction to fall off, or at least move out of the way. If the problem is environmental (heat, humidity, etc.), disks will sometimes recover when left in a cool (not cold), dry place for a while.

Overall prevention: Treat disks with more care—the fragility of magnetic media is underrated. Use the protective jacket *whenever* the disk is not in the machine. Don't lay a disk on top of the video display or near a line printer; these produce magnetic fields which could play havoc with your disks.

Second, simple but devastating: you've changed disks and forgotten to use the I command to initialize it. This needs to be done because much of the disk directory is stored in RAM. The directory is more important than it looks; it contains vital infor-

```
Note: Row signifies vertical (Y) position and column horizontal (X)

For normal video (white on black) the code is as follows:

SET:    PRINT @(row,column),CHR$(26);CHR$(32);CHR$(25);
RESET:  PRINT @(row,column),CHR$(32);

For reverse video, use this code:

SET:    PRINT @(row,column)CHR$(25);CHR$(32);CHR$(26);
RESET:  PRINT @(row,column)CHR$(32);
```

Fig. 1. BASIC Code for SET and RESET

DISK NAME: TRSDOS   DRIVE:0   00/00/00   00.04.13

FILE NAME	CREATED MM	DD	YY	ATTRB	FILE TYPE	REC LEN	NMBR RECS	NMBR EXTS	SPACE ALLOC	USED	EOF BYTE
PAYROLL/TMP	0	0	0	D*X0	F	256	23	2	25	23	0
PAYBAS/TMP	0	0	0	D*X0	F	1	****	1	25	***	***
PAYDATA/TMP	0	0	0	D*X0	F	256	23	1	25	23	0
DOCOM64	11	17	79	D*X0	F	1	167	1	5	1	166
BASCOM64	11	17	79	D*X0	F	256	4	1	5	4	0
COMSUB64	11	17	79	P*X0	F	256	1	1	5	1	0
DOCOM32	11	17	79	D*X0	F	1	167	1	5	1	166
BASCOM32	11	17	79	D*X0	F	256	4	1	5	4	0
COMSUB32	11	17	79	P*X0	F	256	1	1	5	1	0
D V N	4	7	80	D*X0	F	256	1	1	5	1	0
G V	3	3	80	D*X0	F	256	4	1	5	4	0
	0	0	0	D*B0	F	256	****	0	0	***	***
DATM64	11	17	79	P*X0	F	256	4	1	5	4	0
EXDATM64	11	17	79	P*X0	F	256	2	1	5	2	0
DATM32	11	17	79	P*X0	F	256	4	1	5	4	0
EXDATM32	11	17	79	P*X0	F	256	2	1	5	2	0
HERZ50	11	28	79	D*X0	F	1	569	1	5	3	56
	0	0	0	D*B0	F	256	****	0	0	***	***
	0	0	0	D*B0	F	256	****	0	0	***	***
	0	0	0	D*B0	F	256	1	0	0	1	0
	0	0	0	D*B0	F	256	1	0	0	1	0
	0	0	0	D*B0	F	256	1	0	0	1	0
	0	0	0	D*B0	F	256	1	0	0	1	0
	0	0	0	D*B0	F	256	1	0	0	1	0
	0	0	0	D*B0	F	256	1	0	0	1	0
	0	0	0	D*B0	F	256	1	0	0	1	0
	0	0	0	D*B0	F	256	1	0	0	1	0
	0	0	0	D*B0	F	256	1	0	0	1	0
	0	0	0	D*B0	F	256	1	0	0	1	0
	0	0	0	D*B0	F	256	1	0	0	1	0

Fig. 2. Directory of Uninitialized Diskette

mation about the location of each file, and what locations are available. If you change disks and don't initialize, the operating system tries reading and writing to the new disk, using the directory from the old one. This results in areas on the disk being destroyed, and blank file names often show up in a directory as a signal of this (Fig. 2). If the directory looks like this, try typing I (or SYSTEM "I" from BASIC). This will work if you haven't written to the disk (SAVE, PRINT#1, PUT, etc.). But if a directory taken after initialization reveals blank file names, it's too late. Prevention: make it a habit to use the I command before your disk write operations. It doesn't take long, and SYSTEM "I": SAVE"FILE/TXT" is pretty easy to use. SYSTEM "I" also makes a good first program line, if your program uses disk I/O.

Third, a disk is left in the drive after the power is shut off. The heads may release some stored energy at this time, and guess where it ends up? On your disk. The only solution is prevention —*don't* leave a disk in any drive when you power down (or up, just to be on the safe side). This is an unstable state for any electronic device.

Q: I have a BASIC program from my old Model I that uses PEEK and POKE, and I want to put it on my new Model II. Can I?

A: Yes, in most cases. Before

61440	F000	D5	PUSH DE
61441	F001	E1	POP HL
61442	F002	46	LD B, (HL)
61443	F003	23	INC HL
61444	F004	5E	LD E, (HL)
61445	F005	23	INC HL
61446	F006	56	LD D, (HL)
61447	F007	D5	PUSH DE
61448	F008	E1	POP HL
61449	F009	5E	LD E, (HL)
61450	F00A	23	INC HL
61451	F00B	56	LD D, (HL)
61452	F00C	2B	DEC HL
61453	F00D	EB	EX DE,HL
61454	F00E	7E	LD A, (HL)
61455	F00F	EB	EX DE,HL
61456	F010	77	LD (HL),A
61457	F011	C9	RET
61458	F012	00	NOP
61459	F013	00	NOP
61460	F014	00	NOP

To save, type DUMP PEEK/LOD START = F000,END = F011,RORT = R

Fig. 3. Code for PEEK User Routine

61488	F030	D5	PUSH DE
61489	F031	E1	POP HL
61490	F032	46	LD B, (HL)
61491	F033	23	INC HL
61492	F034	5E	LD E, (HL)
61493	F035	23	INC HL
61494	F036	56	LD D, (HL)
61495	F037	D5	PUSH DE
61496	F038	E1	POP HL
61497	F039	5E	LD E, (HL)
61498	F03A	23	INC HL
61499	F03B	56	LD D, (HL)
61500	F03C	23	INC HL
61501	F03D	7E	LD A, (HL)
61502	F03E	EB	EX DE,HL
61503	F03F	77	LD (HL),A
61504	F040	C9	RET
61505	F041	00	NOP
61506	F042	00	NOP
61507	F043	00	NOP
61508	F044	00	NOP

To save, type DUMP POKE/LOD START = F030,END = F040,RORT = R

Fig. 4. Code for POKE User Routine

SUPER DISCOUNTS ON ALL COMPUTER EQUIPMENT

16K MEMORY .... ONLY $25.95! For **TRS-80** Keyboard or Expansion interface. KEYBOARD requires jumpers: $2.00 Extra. These are 200 ns tested RAM for the **TRS-80, APPLE** or **EXIDY.**

DISK DRIVES for the TRS-80 OR PMC-80:

All of our drives come complete with power supply and chassis. They may be used with existing Radio Shack drives on the same cable! 40 track drives store 102K bytes single density, and 175K double density. 80 track drives have 175K single density and 345K double density! All drives guaranteed 90 days, one year on power supply.

40 track MPI drives $319.95
40 track TEAC drives $315.95
40 track TANDON drives $319.95
80 track MPI drives $449.95
80 track TEAC drives $429.95
2 drive cable .... $ 25.95
4 drive cable .... $ 39.95
NEWDOS 80 OPERATING SYSTEM $139.95
NEWDOS 80 PATCH Patches **NEWDOS 80** to work with single or double density and the doubler................................ $ 59.95
PERCOM'S DOUBLER for double density operation! $219.95
The **DOUBLER** works with the **TRS** or **PMC** expansion interfaces to allow you to use your drives in double density! You may still operate your drives as single density also! Comes with **DBLDOS** operating system which allows you to transfer single density files to double and vice versa! *GREAT BUY!*
DISKETTES: VERBATIM DATALIFE! *BOX OF TEN* SOFT OR HARD SECTORED 5¼" $32.50

WE HAVE DRIVES AND CONTROLLERS FOR THE MODEL III. CALL FOR PRICES!!!

MODEMS AND TELE-COMMUNICATIONS

LYNX Telecommunication system for the TRS-80 and PMC-80 ....$279.95
Includes terminal software and connections for instant hookup! Can be connected to the **TRS-80** or **PMC-80** with or without an expansion interface!

LEXICOM MODEM 300 BAUD Requires RS-232.................................. $169.95
THE SOURCE: Hook-up to the "SOURCE" .... $ 99.95
ATARICONNECTION: Modem for 400/800, complete with software! $249.00
APPLECONNECTION: Modem for APPLE II $279.95

COMPLETE SYSTEMS:

PMC-80, 16K LEVEL II COMPUTER $739.00
The **PMC-80** is a work alike to the **TRS-80** mod I computer! Comes with Microsoft's **BASIC** in ROM. Built in cassette. 12" green screen video monitor. Expandable to 48K.
Compatible to All **TRS-80 MOD I Programs.**

PMC 80 without monitor $595.00
RF—MOD for PMC to TV hookup $39.95
PMC-80 EXPANDER 100 SYSTEM $644.00
INCLUDES: 32K memory, S-100 bus, RS-232 interface, Parallel printer driver, Disk controller. Fully compatable with **TRSDOS, NEWDOS, VTOS**, and all other **TRS-80 Mod I disk software!**
APPLE II COMPUTERS 16K $1195.00
APPLE II COMPUTERS 48K $1299.00
ATARI 400 .... $ 495.95
ATARI 800 COMPUTER .... $ 795.00
ZENITH Z-89 48K, 1 DISK ALL IN ONE COMPUTER $2495.00

VIDEO MONITORS

LEEDEX 100 12' B/W MONITOR $139.95
SANYO 9' B/W MONITOR $199.95

PRINTERS

OKIDATA MICROLINE 80 All OKI'S have TRS-80 Graphics!!! .... $ 499.00
Comes with friction and pin feed, upper/lower case, Graphics.
EPSON MX-80 PRINTER: Word Processing Quality Printout with Graphics!..................... Call
NEC SPINWRITER with Tractor Feed ... $2995.00
DIABLO MODEL 1630 with Tractor $2695.00
UNIVERSAL PRINTER STANDS .... $ 94.50

Other Accessories:

SUP-R-MOD RF Modulator for APPLE ... $29.95
ARCHBOLD SPEED-UP MOD FOR TRS MOD I. Allows up to 300% increase! $45.00
MICROSOFT BASIC Decoded and other mysteries $29.95
TRS-80 DISK and other mysteries $18.95
ZBASIC BASIC COMPILER for MOD I and MOD III TRS-80 and PMC-80 Increase basic program speeds by up to 200 times!!!!!
Tape .... **$79.95** **Disk** **$89.95**
Both For Only $99.95 Specify MOD I or MOD III

WE HAVE HUNDREDS OF PROGRAMS FOR ALL THE POPULAR COMPUTERS! CALL FOR OUR FREE CATALOG

Personal and Business Checks take 3 weeks to clear. For fastest delivery use Certified Checks, Money Orders, Credit Card, or C.O.D

We will try and beat any published price on any system! Call!

SIMUTEK COMPUTER PRODUCTS ™ ✓19

4877 E. Speedway Blvd.
Tucson, Arizona 85712
(602) 323-9391 Technical Questions

CALL TOLL FREE TO ORDER
(800) 528-1149
Please mention this magazine when ordering
ARIZONA RESIDENTS ADD 6% SALES TAX.
WE TAKE VISA, MASTER CARD.

✓See List of Advertisers on page 306

you upgrade, though, you have to find out why PEEK and/or POKE are used, and see if there is really a need. If, for instance, POKE graphics are used to write to screen memory, you'd better stick with PRINT @ statements, as screen memory is only accessible through a supervisor call on Model II. If, however, you find that you *must* use PEEK or POKE, use the code in Figs. 3 and 4 (see Fig. 5 for information on entering machine language programs). Use the accompanying BASIC program in Fig. 6.

Q: I just jumped from BASIC to DOS using System. Is there any way to go back to BASIC without losing my program?

A: Enter the program in Fig. 7 (again, following the guidelines in Fig. 5 for entry and storage). As long as you returned from BASIC and didn't use any DOS commands that dump or zero memory, the return will always be successful.

To load the code into memory:
From TRSDOS READY, type
DEBUG ON (enter)
DEBUG (enter)

When the "?" appears on the screen, reply with "M" followed by the starting address of the code. Tap the F1 key to position the cursor for entry of the code. Enter the code as it appears, in hex. When entry is complete, tap the F2 key to store it in memory, then "S" to return you to TRSDOS READY mode.

To store the program, type
DUMP name START = address1, END = address2, RORT = X (enter) where "name," "address1," "address2," and "x" are replaced with the values given for each program. So, to save the J2800 program, type
DUMP J2800 START = F100, END = F102, RORT = T (enter)

To call the program from TRSDOS READY, type name (enter) where name is the program name used in DUMP. From BASIC, use SYSTEM "name".

Fig. 5. Loading and Saving a Machine Language Program

The following BASIC code will make PEEK and POKE easier to use in your programs:

To load PEEK and POKE into memory and set up the USR routines–

```
60000 SYSTEM"PEEK/LOD":SYSTEM"POKE/LOD":DEFUSR1 = &HF000:
DEFUSR2 = &HF030
```

To convert a memory address into integer format (which must be done before each PEEK or POKE call) where address is contained in X and returned as an integer in A%:

```
65000 IF X> = 32768 THEN A% = ( - 32768) + (X - 32768):ELSE A% = X
65010 RETURN
```

To perform a PEEK (read from memory address X):

```
65100 GOSUB 65000'Address is already in X - Convert to integer A%
65110 A$ = MKI$(A%)'Prepare address
65120 A$ = USR1(A$)'Call PEEK
65130 B% = ASC(A$)'B% now contains value of memory address X
```

To perform a POKE (place value V% at memory location X):

```
65200 GOSUB 65000'Convert X to integer A%
65210 A$ = MKI$(A%)'Prepare address
65220 A$ = A$ + CHR$(V%)'Tack on value to store at X
65230 'NOTE: Value (V%) must be between 0 and 255, inclusive
65240 A$ = USR2(A$)'Call POKE - V% is now stored at memory location X
```

Fig. 6. BASIC Code for use with PEEK and POKE

61952	F200	C3<0028>	JP <NN> * TO:2800
61955	F203	00	NOP
61956	F204	00	NOP
61957	F205	00	NOP

To save, type DUMP J2800 START = F200,END = F202,RORT = T

Fig. 7. Code for J2800 user routine.

Q: I have a machine language program from the Model I which contains various calls to ROM routines, for routines such as keyboard input and disk I/O—Is there an equivalent on the Model II?

A: Yes, there is. The routines are named Supervisor Calls (SVCs). They are called by loading the arguments into the proper registers and executing an RST 8 instruction.

A list of commonly used routines is in Fig. 8, and the DOS manual covers them nicely in pages 4/13 to 4/84.

Q: I have a program in BASIC which must be secured from Break. Can I disable the Break key temporarily?

A: Yes. The codes in Figs. 9 and 10 will permit you to disable, or enable, the Break key. Be careful to use them only inside BASIC code. Remember to enable the Break key before program execution is over.■

SVC CODE	Description
15	Read the disk ID from any drive
25	Set a timer to generate an interrupt after n seconds
36	Jump to TRSDOS READY mode
38	Execute a DOS command
4	Fetch a character from the keyboard
5	Fetch a line from the keyboard
7	Clear the screen with normal/reverse video
8	Output a character to the display
9	Output a line to the display
11	Read video memory
18	Send a character to the printer
19	Send a line to the printer
40	Open a disk file
35	Read from a disk file
44	Write to a disk file
42	Close a disk file
41	Delete a disk file
20	Generate a random number
21	Perform binary-decimal/decimal-binary conversions
23	Multiply/divide 16 bits by 8 bits
24	Perform binary-hex/hex-binary conversions

Fig. 8. Sample supervisor calls.

```
61584   F090   3E<03>     LD A,<N> * <03>
61586   F092   21<0000>   LD HL,<NN> * <0000>
61589   F095   CF         RST 8
61590   F096   3E<03>     LD A,<N> * <03>
61592   F098   21<9CF0>   LD HL,<NN> * <F09C>
61595   F09B   CF         RST 8
61596   F09C   C9         RET
61597   F09D   00         NOP
61598   F09E   00         NOP
61599   F09F   00         NOP

To save, type DUMP DISABLE/BRK START=F090,END=F09C,RORT=T
```

Fig. 9. Code for DISABLE/BRK user routine.

```
61584   F090   3E<03>     LD A,<N> * <03>
61586   F092   21<0000>   LD HL,<NN> * <0000>
61589   F095   CF         RST 8
61590   F096   3E<03>     LD A,<N> * <03>
61592   F098   21<0260>   LD HL,<NN> * <6002>
61595   F09B   CF         RST 8
61596   F09C   C9         RET
61597   F09D   00         NOP
61598   F09E   00         NOP
61599   F09F   00         NOP
61600   F0A0   00         NOP

To save, type DUMP ENABLE/BRK START=F090,END=F09C,RORT=T
```

Fig. 10. Code for ENABLE/BRK user routine.

✓See List of Advertisers on page 306

A detailed examination of the design, logic and capabilities of Tandy's exciting new machine.

The Color Computer

Phillip Martel
748 Tyler St.
Pittsfield, MA 01201

Robert Nicholas
2-8 Lennox Heights
Lenox, MA 01240

So—your friendly Radio Shack store is selling *color* computers; and high-resolution color at that. No more 48 by 128 white blocks, but as many as eight colors (nine if you count black) in five modes ranging from 32 × 64 to 192 × 256.

Why would anyone want to buy a color computer from Radio Shack when you can get an Apple for only $1195 or an Atari for about $1079? The most convincing sales pitch is the price—$399! That gets you a 4K computer with low-level Color BASIC. You can move up to 16K for another $119 (plus installation). The total 16K computer with extended BASIC costs only $599.

What Comes with It?

For the initial $399, you get the following:

- 4K Color BASIC computer with built-in RS-232 I/O port and real-time clock
- TV switch box
- 12 foot cable to connect computer to switch box
- Operation Manual
- Color BASIC instruction manual
- Card listing statements, functions, operators, etc.

Phil's system arrived minus the TV switch box and card. Included with the system is a form to return, stating that more information will be made available in the future. You will also be placed on the Radio Shack computer newsletter mailing list.

You must supply your own color (or black and white) television set and tape recorder. The TRS-80 Color Video Receiver (#26-3010) is a 13 inch TV set that sells for $399. The CTR-80A Cassette Recorder (#26-1206) costs $59.95. Floppy disks are supposed to be available in the future. In the meantime, don't despair; the cassette interface operates at 1500 baud—three times the speed of Level II BASIC and six times faster than Level I!

The computer uses a 6809E Motorola eight-bit microprocessor chip with a clock speed of 0.894 MHz. There are DIN connections at the back of the computer for a tape recorder, serial printer or modem, and dual joystick controllers (#26-3008, $24.95). There is also a switch to select channel three or four for computer operation.

Hooking It Up

The operation manual clearly describes how to hook the TV switch box to your set. After plugging the computer into the wall and connecting the computer to the switch box with the cable provided, flip the switch from the TV setting to the computer setting. Now turn on your television to channel three or four, whichever proves to be clearest, and press the on/off button on the computer.

You should now have a green screen framed in black with the Tandy message in the upper left corner. The operation manual includes a color adjustment test and a video centering test to help you balance your set for computer use.

The clarity and color quality of your screen will depend upon the condition of your television set. We are using a ten-year-old Zenith with one and a new Samsung with the other. We have also seen the Shack's video monitor working. While all three sets provide easily readable text, some of the colors are not very sharp. The worst cases are red, yellow and orange. Red lines particularly appear to be different shades, depending upon whether the line is horizontal, vertical or diagonal. The resolution is poor on the old Zenith, better on the new Samsung and better yet on the Shack's monitor.

At this point, you should have 2343 bytes of available memory with a 4K machine and 14631 bytes with 16K. Two hundred additional bytes have been automatically set aside for strings.

If you have one of the old Radio Shack cables for a tape recorder, be advised that the color computer requires a *metal* DIN plug. The plastic ones won't fit in the slot provided.

Color BASIC and the Manual

Color BASIC has more power than Level I BASIC, but less than Level II. A complete list of commands is given in Table 1. In addition, there are the usual symbols for addition, subtraction, multiplication and division. You cannot use exponents. The relational tests of <, >, =, < =, > = and <> exist, as do the logical operators NOT, AND and OR.

Permitted abbreviations are the apostrophe for REM and question mark for PRINT. Multiple lines are supported using the colon as a separator. The comma and semicolon are used with PRINT to control spacing. A comma provides 16-column zones. The video display is 16 lines of 32 characters each. Shift @ halts listing or program flow, as on the Model I.

Variables may be one or two characters long. (They may actually be longer, but only the first two are significant.) Numeric and string variables, as well as arrays, may be used. The LET command has been completely obliterated. If you enter LETA = 25 and then ask for the value of A, you will get a zero. However, the variable LETA has a value of 25.

Some of the commands are different from any in either Level I or II. The CLEAR command allows you to clear string space

"Color BASIC has more power than Level I BASIC, but less than Level II."

and set memory size at the same time. (There is no more memory size question.) For instance, CLEAR500,12000 would reserve 500 bytes of string storage and set memory size at 12000.

CLS does what it used to do—sort of! It now clears the screen to its usual green color. You can change this by specifying a number from 0 to 8. CLS4 clears the video screen to red. Color numbers are:

0 black	3 blue	6 cyan
1 green	4 red	7 magenta
2 yellow	5 white	8 orange

Entering CLS9 gives a green screen with *Microsoft* written in the upper left corner. This is true of all numbers from 9 to 255.

There is no CLOAD?, which means you cannot verify programs saved to tape. To make up for it, there's Audio. Entering AUDIO ON before a CLOAD or CSAVE will project your program over your TV speaker. AUDIO OFF turns the sound off. You can also save programs to tape in ASCII format. Example: CSAVE"MEMORY",A.

Machine language programs load with the CLOADM command rather than SYSTEM. EXEC is used to transfer control to the machine language program. You can specify an address, such as EXEC15348. (You also have USR for transferring control to machine language routines.)

You can OPEN sequential files to the screen or keyboard (0), cassette (−1) or line printer (−2). PRINT# and INPUT# are used for transferring information to and from the file. There is also an EOF (end of file) feature.

You can LLIST an entire program or any part thereof. LPRINT, however, has been replaced by PRINT#−2. We have experienced some problems with LLIST. It seems to scramble lines once in awhile. (If you have a copy of *80 Programs for the TRS-80* from *80 Microcomputing*, you can modify the print program on page 194 and list and format your program saved on tape in ASCII format.)

A motor feature has been added that turns the recorder on or off from the keyboard without pulling plugs. (Hint: Rather than typing MOTOR OFF, enter anything which would be an error—it's faster. Example: Type H and then enter.)

Programs can be saved to tape with names up to eight characters in length. SKIPF allows you to move to the end of the next file. SKIPF"MEMORY" would search for the file named MEMORY and stop at the end of it.

The SET command includes a color element. SET(13,24,4) would light up block (13,24) with the color four (red). RESET does not need a color indicator.

ABS	ASC	AUDIO	CHR$
CLEAR	CLOAD	CLOADM	CLOSE
CLS	CONT	CSAVE	DATA
DIM	END	EOF	EXEC
FOR...TO...STEP/NEXT		GOSUB	GOTO
IF...THEN...ELSE		INKEY$	INPUT
INPUT#	INT	JOYSTK	LEFT$
LEN	LIST	LLIST	MEM
MID$	MOTOR	NEW	ON...GOSUB
ON...GOTO	OPEN	PEEK	POINT
POKE	PRINT	PRINT@	PRINT#
PRINTTAB	READ	REM	RESET
RESTORE	RETURN	RIGHT$	RND
RUN	SET	SGN	SIN
SKIPF	SOUND	STOP	STR$
USR	VAL		

Table 1. TRS-80 Color BASIC Statements and Functions

JOYSTK finds the horizontal and vertical screen positions of the dual joysticks. Use memory location 65280 to determine if a joystick button has been depressed.

Sound fanatics; we now have SOUND. You can send a frequency from 1 (lowest pitch) to 255 (highest) through your TV speaker for a duration of 1 to 255, making programming games and error detection much easier.

Twenty-five error messages are given in two or three character codes.

The BASIC manual covers most commands. It assumes you know nothing about programming and uses the tongue-in-cheek humorous approach of the original Level I manual. No mention is made of file handling. The basics of numbers, strings, graphics, color, joysticks and sound are covered adequately. The manual refers the user to other computer programming books available from Radio Shack for "more complete instructions on how to program."

The appendices provide a table of values for musical tones with SOUND, a PRINT@ screen location table and a table for SET locations. There are also a few sample programs.

Table 2

Table 2 is a memory map of the Color Computer. It is similar to the memory map in Appendix D of Radio Shack's Level II manual.

Unlike the Model I, the Color Computer has its user RAM (Random Access Memory) in low memory and its ROM (Read Only Memory) higher up. This is because of the different ways the Z-80 chip and the 6809 chip behave on power up and restart.

The Z-80 starts executing code at address 0, so ROM must be in low memory. The 6809 starts looking at the two locations at top of memory to determine the starting location. This means that some ROM must be in the top of memory.

The Color Computer has a 32 byte ROM in locations FEOO to FFFF hexadecimal. This ROM can be disabled by inserting a ROM cartridge in the slot located on the side of the computer. This approach allows the ROM cartridge programs to use some of

Hexadecimal	Decimal	Contents
0000-3FFF	0 -16383	User RAM (16K)
4000-7FFF	16384-32767	Empty (RAM expansion?)
8000-9FFF	32768-40959	Who knows what evil lurks?
A000-BFFF	40960-49151	8K Color BASIC ROM
COO-DFFF	49152-57343	Memory location for program pack ROMs
E000-FEFF	57344-65279	Who knows what evil lurks?
FF00-FF1F	65280-65311	Keyboard and joystick button PIA (input/output)
FF20-FF3F	65312-65343	Cassette, joystick and video PIA (input/output)
FF40-FFDF	65344-65503	Part of this looks like control for high resolution
FFE0-FFFF	65504-65535	Transfer vector ROM

Table 2. Overall Memory Map

"The BASIC manual. . . assumes you know nothing about programming and uses the tongue-in-cheek humorous approach. . . ."

BASIC Keyword	Stored Decimal	Value Hexadecimal	ROM Address Decimal	ROM Address Hexadecimal
FOR	128	80	44359	AD47
GO	129	81	44678	AE86
REM	130	82	44771	AEE3
'	131	83	44771	AEE3
ELSE	(58),132	(3A),84	44771	AEE3
IF	133	85	44820	AF14
DATA	134	86	44768	AEE0
PRINT	135	87	47351	B8F7
ON	136	88	44866	AF42
INPUT	137	89	45045	AFF5
END	138	8A	44546	AE02
NEXT	139	8B	45304	B0F8
DIM	140	8C	45902	B34E
READ	141	8D	45126	B046
RUN	142	8E	44661	AE75
RESTORE	143	8F	44516	ADE4
RETURN	144	90	44736	AEC0
STOP	145	91	44553	AE09
POKE	146	92	46935	B757
CONT	147	93	44592	AE30
LIST	148	94	46948	B764
CLEAR	149	95	44609	AE41
NEW	150	96	44311	AD17
CLOAD	151	97	42136	A498
CSAVE	152	98	42060	A44C
OPEN	153	99	42486	A5F6
CLOSE	154	9A	42006	A416
LLIST	155	9B	46942	B75E
SET	156	9C	43136	A880
RESET	157	9D	43185	A8B1
CLS	158	9E	43280	A910
MOTOR	159	9F	42941	A7BD
SOUND	160	A0	43339	A94B
AUDIO	161	A1	43408	A990
EXEC	162	A2	42302	A53E
SKIPF	163	A3	42476	A5EC
TAB(	164	A4		
TO	165	A5		
SUB	166	A6		
THEN	167	A7		
NOT	166	A8		
STEP	169	A9		
OFF	170	AA		
+	171	AB		
–	172	AC		
*	173	AD		
/	174	AE		
↑	175	AF		
AND	176	B0		
OR	177	B1		
>	178	B2		
=	179	B3		
<	180	B4		

Table 3. One-byte BASIC Keyword Codes and Addresses

the BASIC ROM's subroutines.

The sections of memory space marked as empty have the value 255 (hexadecimal FF) in them. The sections of memory space (Table 2) marked "Who knows what evil lurks?" have the value 126 (hexadecimal 7E) in them, as does the section of memory space marked for high resolution. What goes on in this space is unknown to us at this time.

The two sections of memory described as PIA in the table are I/O (input/output) devices. There are I/O ports built into the Z-80. The 6809 lacks these ports and does all its I/O through the memory. The concept of memory mapped I/O is described in the *TRS-80 Microcomputer Technical Reference Handbook* (R.S. Cat. No. 26-2103). These locations control all the keyboard, cassette, and joystick interactions and some of the video generator controls.

The joystick buttons are brought in through the keyboard port. In addition to PEEKing location 65280, these buttons may be detected by an INKEY$ statement.

Table 3 contains a list of the BASIC words stored in the program as one byte. This data is available in ROM starting at location 43622 (AA66 hexadecimal). If you examine the ROM, you will notice the last letter of the word is garbled. In fact the last letter has had 128 (80 hexadecimal) added to it as an end of word indicator.

The stored value is the actual code stored in the BASIC program's memory space. It corresponds to the value 128 plus the position of the BASIC keyword in the table. The instructions FOR to SKIPF (with the exception of ELSE) may start a line of the BASIC program.

A table in ROM, starting at location 43879 (AB67 hexadecimal), contains the starting addresses of the routines that perform the keywords. Table 3 lists these values in the fourth and fifth columns. We haven't been able to locate the addresses for those keywords from TAB(down.

BASIC Keyword	Stored Decimal	Value Hexadecimal
SGN	128	80
INT	129	81
ABS	130	82
USR	131	83
RND	132	84
SIN	133	85
PEEK	134	86
LEN	135	87
STR$	136	88
VAL	137	89
ASC	138	8A
CHR$	139	8B
EOF	140	8C
JOYSTK	141	8D
LEFT$	142	8E
RIGHT$	143	8F
MID$	144	90
POINT	145	91
INKEY$	146	92
MEM	147	93

NOTE: Each of these keywords actually uses two bytes. The first byte is 255 (or FF hexadecimal). The second byte is as indicated above.

Table 4. Two-byte BASIC Keyword Codes

A word is in order about ELSE. In the BASIC program, ELSE is stored as a 132 (84 hexadecimal), as indicated. However, a colon (58 decimal, 3A hexadecimal) is always stored in front of it. (This colon is stored in memory. It doesn't appear in the program listing.) If you remove the colon by POKEing the location which contains it with another value, you can get a program which looks correct but which gives a syntax error when the THEN branch of the IF. . .THEN. . . ELSE statement is run.

Table 4 lists keywords which are preceded by 255 (FF hexadecimal) in the BASIC program memory. In ROM this table follows the other keyword table. It starts at 48802

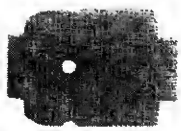
TRS-80® DISCOUNT $ SAVE $

MODEL III

26-1061 4K I	$599.00
26-1062 16K III	859.00
26-1063 32K III W/2 Drives, RS 232	2,225.00
26-1162 1st Drive	765.00

Color Computer 4K $310

26-3001 4K	$353.00
26-3002 16K Ext. Basic	533.00
26-3009 Joysticks	22.50
26-3010 Color Video	353.00
26-1206 Recorder	54.00



Proto Board For Color Computer —
Fits inside 8-Track Cartridge **$24.95**

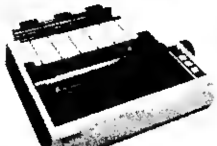
Model I

26-1140 Expansion Interface	$249.00
26-1141 16K Exp. Interface	359.00
26-1142 32K Exp. Interface	469.00
26-1145 RS 232C Board	84.00
26-1160/1 Mini Disk Drive	419.00
26-1563 Scripsit-Disk	79.00
26-1566 Visicalc	83.00

MODEL II

Model II 64K $3395

26-4160 1 Drive Exp	$1014.00
26-4161 2 Drive Exp	1559.00
26-4162 3 Drive Exp	2099.00
26-4530 Scripsit II	259.00
26-4512 Profile II	159.00
26-4511 Visicalc II	259.00
26-4501 Gen Ledger	179.00
26-4506 Mail List	72.00



PRINTERS

26-1167 9½ Line Printer VII	$ 360.00
26-1166 Line Printer VI	1080.00
26-1158 Daisy Wheel II	1798.00
26-1165 Line Printer V	1710.00
26-1401 Cable	36.00

EPSON MX80/MX70

Low-Priced Professional Print Quality

EPSON MX80	(List $645)	$499.00
EPSON MX70 Dot Graphics, 5x7 Matrix	(List $450)	$425.00

IDS PAPER TIGERS

Dot Resolution Graphics quality print

IDS 445G 7 wire printhead, graphics	(List $985)	$ 795.00
IDS 460G 9 wire printhead, graphics	(List $1394)	1195.00
IDS 560G 9 wire, wide carriage, graphics	(List $1794)	1529.00

No Taxes on Out Of State Shipments

Okla Include 2% State Tax

All prices are subject to change without notice. Freight, handling, and insurance charges are extra. Most items are shipped United Parcel Service. Model II and other hardware weighting over 50 lbs. is shipped freight collect by truck. Certified Check for immediate shipment from stock. Master Charge, Visa or Bank Card add 3% surcharge.

DOES YOUR SMART PRINTER SUDDENLY BECOME DUMB WHEN YOU PRINT FROM SCRIPSIT?

Is your printer capable of underlining but not from Scripsit?

Is your printer capable of **Bold Printing** but not from Scripsit?

Can your printer super script and sub script but not from Scripsit?

Can your printer change pitches but not from Scripsit?

The answer is SCRIPMOD

SCRIPMOD does not require a separate printer driver. With SCRIPMOD control codes can be embedded in the test of your document. You use the same format line syntax you're used to now. SCRIPMOD adds one format instruction to Scripsit & two control codes.

SCRIPMOD is supplied on disk with full documentation for **$39.95.**

PENCIL-FIX Save your warranty. Use PENCIL-FIX to avoid custom control key on your keyboard. Redefines the control key for EP to be the @ key Use RS lower case mod or the EP mod without the control key. Disk based EP only . . **$14.95**

PRINT-CENTRAL Send any control code directly to your smart printer from the BASIC command mode or from DOS Avoid having to type such things as "LPRINT CHR$(31) just to change pitch. Use CLEAR right arrow (two key strokes) instead Any code from 1 to 31 may be sent. Mod I 32K disk . . **$24.95**

SPOOL-REL An in-memory print buffer that runs in Mod I 32K or 48K disk systems. Fully relocatable code and buffer. A true background spooler at an unbelievably low price . **$24.95**

TIGGER-GRAF Create engineering, scientific, business, or just plain fun on your IDS 440G or 460G printer. Resolution is 495 x 575. Easy BASIC programs provided for data entry and machine language module for speed. Includes setting individual points, drawing lines, shading shapes. Several graphs may be catenated along the Y-axis for larger graphs. Requires Mod I 32 or 48K 1 disk . . . **$149.95**

ALL POCKET AND COLOR COMPUTER SOFTWARE SOLD AT DISCOUNT

WRITE US FOR A FREE CATALOG

1-800-331-9128 Toll Free Order Entry

Immediate Shipment From Stock on Most Items

Vern Street Products ✓111
The Computer Store, Inc.

114 West Taft
Sapulpa, Oklahoma 74066
918-747-9333

4949 South Peoria
Tulsa, Oklahoma 74105

TRS-80 is a registered trademark of the Tandy Corp.

✓See List of Advertisers on page 306

"Unlike the Model I, the Color Computer has its user RAM in low memory and its ROM higher up."

Memory Location	191	223	239	247	251	253	254
338	ENTER	8	0	X	P	H	@
339	CLEAR	9	1	Y	Q	I	A
340		:	2	Z	R	J	B
341		;	3	↑	S	K	C
342		,	4	↓	T	L	D
343		—	5	←	U	M	E
344		.	6	→	V	N	F
345		/	7	SPACE	W	O	G

Note: Normal value at these locations is 255. When a key has been pressed, the value changes accordingly. Shifted keys produce the same values. Note also that the values above are the 255 complements of 64, 32, 16, 8, 4, 2 and 1.

Table 5. TRS-80 Color Computer Keyboard Map

(AB1A hexadecimal) and runs to 43878 (AB66 hexadecimal). The table of starting addresses displayed in Table 3 comes after this table of keywords. Again, we have not yet been able to find the ROM start addresses for these keywords.

Table 7 lists the locations of functions in RAM that we have found. In the 6809, the memory may be considered to be divided into 256 pages of 256 bytes each. An internal register, the DPR (direct page register), points to one of these pages. This allows quick access to a page. In much, if not all, of Color BASIC the DPR points at page 00, which holds quite a few of the pointers and storage variables used by BASIC.

One potentially useful section of page 01 is found in Table 5, containing a memory map of the keyboard (338 to 345 decimal). All keys except the Break and Shift keys can be accessed from this map. Through careful programming you can determine if a particular key is being depressed. For instance, if PEEK(344) equals 247 then the right arrow key is being held down. When you release the key, memory location 344 will revert to 255.

Another handy location is 135 (87 hexadecimal), which normally contains a zero, and holds the ASCII value of the last key pressed. It will continue to hold that value until you press another key or until either an INKEY$ or INPUT statement is encountered.

The beginning of page 01 (around address 256) holds the starting locations for interrupt routines (SW13, SW12, SW1, NMI, IRQ and FIRQ). These addresses are specified by the transfer vector ROM at the high end of memory. There are three addresses between these locations; normal procedure is to put a jump instruction to the body of the interrupt routine in these locations.

Locations 1024 to 1535 (400 to 5FF hexadecimal) contain the memory normally associated with video display. Unlike the Model I, the Color Computer does not have a physically separate RAM for the video storage. The position and amount of RAM displayed on the screen can be changed by POKEing certain locations in high memory. This process has to be used for some of the higher graphics resolution modes.

Locations 1024 to 1535 (0400 to 05FF hexadecimal) are your video screen. If you want to POKE to the screen or PEEK at it, do it here. According to Table 8, it can make a difference whether you print the CHR$ of a number or POKE that number to the screen. From 0 to 127 you will get control codes, letters, numbers, arrows and the rest of the keyboard symbols. (There is no way to produce the right arrow or the down arrow.) Table 6 shows the lowest resolution graphics codes from 128 to 255. Notice that these blocks repeat in groups of sixteen for each of the eight colors.

Fig. 2 shows how RAM is divided for use by BASIC. The system RAM is used internally by BASIC. The video display RAM is where the screen data is stored. The first address of video RAM can possibly be moved, but we're not certain how to do it. This move involves some POKEing into high memory in the area indicated in Table 2.

The addresses for the rest of the start points are surrounded by parentheses, which indicates that the desired address is stored there. (Actually, it is in that address and the following one. In the 6809, the most significant byte of an address is stored first, while in the Z-80 the least significant byte comes first.)

You can modify these pointers by POKEing the pointer locations. Modifying the pointers is dangerous because you may destroy your program or cause the Color Computer to hang up in a state which it can't get out of without being powered down. However, if you have saved a copy of your program and power down, all will be well. You can't damage the system by typing on the keyboard, so experiment!

And Now, The Programs

The first program is titled Memory. It allows you to search through RAM and ROM to see what's there. You may enter the starting address in decimal or hexadecimal. Your screen will be filled with a block of thirty-two bytes of memory at a time, showing the RAM/ROM location, the number (0-255, OH-FFH) stored there and the CHR$ of that number (if greater than 32). These values can be shown in either decimal or hexadecimal. To switch from decimal to hex, press the H. To return to decimal mode press D. To scroll forward a block of thirty-two bytes press ; and to scroll backwards press –. When you finish looking through one section of code and wish to switch to another, press CLEAR and select a new starting address.

Refer to Table 2 and Table 7 as guides while PEEKing through memory. Note that at 466 RAM (1D2 hex) you will see the name of the program last CSAVEd, CLOADed or SKIPFed, and that the last file encountered on tape will be found to start at 474 (1DA hex). Within the last two hundred bytes of your computer's memory (3895 for 4K and 16183 for 16K), you should begin to see the keys you are pressing while running Mem-

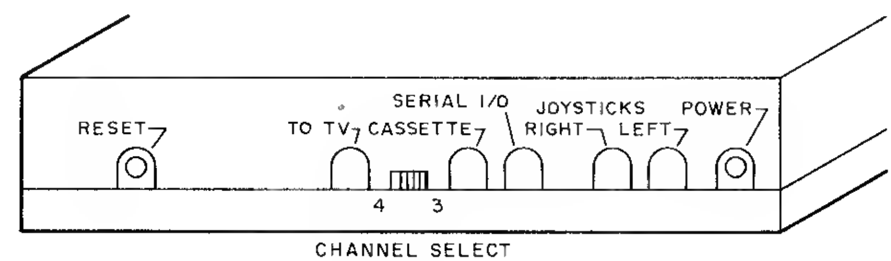


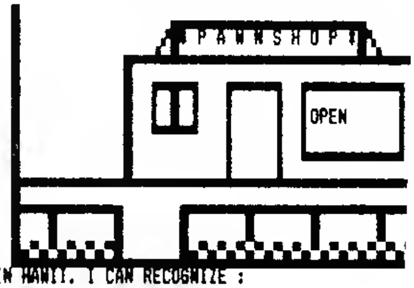
Fig. 1

INTERPRO
presents

TRS-80* ILLUSTRATED ADVENTURE

The Atlantian Odyssey with Graphics

```
OBVIOUS EXITS-
NORTH
```




```
I AM ON A DOCK IN HAWII. I CAN RECOGNIZE :
PAWNSHOP. SAILBOAT.
I HAVE :THE SCUBA GEAR, WHICH I AM WEARING.
* CRYSYAL PYRAMID *. * MEDALLION *.
KNAPSACK CONTAINING: FLASHLIGHT. SPEARGUN.
WHAT DO I DO NOW?
```

MACHINE LANGUAGE

32 Graphic Locations · 150 Word Vocabulary
Exciting Adventure

MODEL I 48K DISK $29.95
WITHOUT GRAPHICS
16K CASSETTE... $14.95

DEALERS INVITED
*TM TANDY CORP

TRS-80* 16 K-up UTILITY

ULTRA-MON

The Intelligent Monitor

The most powerful tool available to the novice or professional machine language programmer.

- ROM/DOS Independent
- Lineprinting disassembler
- Self relocating
- Single step through RAM or ROM with each instruction individually disassembled.

INTERPRETIVE EXECUTION allows you to execute your program or the ROM with Ultra-mon in complete control.

- DEBUG* type display with P.C. register disassembled.
- Breakpoints can be set in RAM or ROM.

.......AND MORE

* EXCELLENT REVIEWS IN APRIL 80 *
* MICROCOMPUTING AND APRIL *
* SOFTSIDE *

MODEL I, **16K CASSETTE**
(Disk Loadable) $24.95

24 PAGE DOCUMENTATION
WITH STEP-BY-STEP
INSTRUCTIONS AND
SIMPLE DEMONSTRATIONS

— *DEALERS INVITED* —
*TM TANDY CORP

TO ORDER
CALL or WRITE
INTERPRO
P.O. BOX 4211 ✓485
MANCHESTER, NH 03108
M-C **(603) 669-0477** VISA

FREE IDEA SEEDS

The most unique concept in software ideas. Are you tired of not knowing those tricks and shortcuts which the expert programmer utilizes without even thinking twice? Now you can pick up some tips and novel routines which will simplify your own BASIC programming. TRS-80 Model I LII.

TRS-80 is a trademark of Tandy Corp.

Free Idea Seeds is a trademark of CECDAT, INC.

All you have to do is send a No. 10 size envelope, self-addressed and stamped, to:

✓62
CECDAT, INC.
P. O. Box 8963
Moscow, ID 83843

Features of the FATIGUE FIGHTER T.M.:

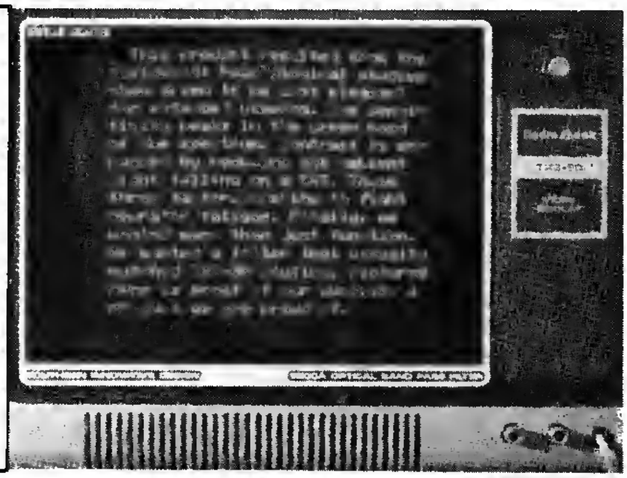
- **REDUCES OPERATOR FATIGUE THEREBY ALLOWING MORE EFFICIENT USE OF THE COMPUTER.**
- **INSTALLS EASILY WITH PRESSURE SENSITIVE ADHESIVE. NO SCREWS, CLIPS, OR DRILLING TO DAMAGE MONITOR.**
- **DESIGNED TO MATCH TRS-80® STYLING FROM THE BLACK AND SILVER BORDER TO THE LETTERING TYPE FACE.**
- **MADE OF TOUGH 1/8 INCH THICK ACRYLIC SHEET FOR DURABILITY, EASE OF CLEANING, AND A QUALITY APPEARANCE.**
- **FITS BOTH THE MODEL I AND THE MODEL II.**
- **DOES NOT VOID THE COMPUTER WARRANTY.**

TO ORDER : Send Name & Address Typed or Clearly Printed with Check or Money Order for $12.95 Each, Including Shipping. COD's are $2.50 Additional Per Order & are Accepted by Mail or by Calling 904-378-2494 9-5 M-F. Florida Residents Add 4% Sales Tax.

TRS-80 IS A TRADEMARK OF TANDY

SiD

SOUTHERN INNOVATIVE DESIGN
1520 NORTHEAST 12TH STREET
GAINESVILLE, FLORIDA 32601 ✓373



"The clarity and color quality of your screen will depend upon the condition of your television set."

0	0000	
		SYSTEM RAM
1024	0400	
		VIDEO DISPLAY RAM
(25)	(0019)	
		BASIC PROGRAM
(27)	(001B)	
		SIMPLE VARIABLE STORAGE
(29)	(001D)	
		ARRAY VARIABLE STORAGE
(31)	(001F)	
		FREE SPACE
(23)	(0017)	
		STACK AREA
(33)	(0021)	
		STRING STORAGE
(39)	(0027)	
		PROTECTED MEMORY
(116)	(0074)	

Fig. 2. Map of BASIC Storage Area

ory (D,H, ;, –).

The list of BASIC keywords runs from 43622 to 43878 (AA66–AB66 hex). The ROM starting addresses for these routines can be found from 43879 to 43950 (AB67–ABAE hex). The twenty-five error messages of Color BASIC can be found from 43951 to 44000 (ABAF–ABE0 hex).

The second program is called Windows. It is a low-resolution graphics program. Format one creates sixteen identical windows on your video screen and format two simultaneously draws four identical Kaleidoscopic patterns. You select the format and the number of points to be drawn. Finally, pick the colors you want used for the points. (The colors will be shown and numbered for your convenience.) You may enter a color more than once to give more emphasis to that color in the design.

Sit back and watch while your computer dazzles your eyes with sparkling graphic displays of random splendor! And listen as well, for it will play along as it draws. When it has finished the design, it will give a little toot and pause shortly before moving on to create yet another pattern. If you wish to change formats or the number of points being used, press the Clear key after you hear the toot and you'll be returned to the beginning of the program to start anew.

Experiment with both formats, the number of points displayed and the color combinations and see what you come up with. Fewer points will give a light airy design, whereas many points will produce more complex and crowded figures.

A Musical Program

The third program listing is Music-Music-Music, and is, obviously, a music machine. There are options to create a song, edit it, add to it, play it, save it to tape and recall a song from tape.

While entering a selection, adding to it or playing the final product, you will see a double-keyboard graphically printed on your video screen with white and black keys. The notes run through four octaves starting with F below middle C. (Middle C is number 8.) While the music is playing, you will notice an orange block bouncing around on each key as that note sounds. This orange block also appears while entering music, to assist you visually.

To enter a note, type in its number (1-48), or zero (0) for a rest. (See Fig. 3.) To cancel the last note entered, type the letter C. To end entering music press the up arrow and hit Enter. (All of this information appears on the screen while you are working.)

Once you've selected the number of the note (or rest), the orange block will appear on that note on the keyboard. Type in the duration for that note (1-32). (See Fig. 4.) If you want to cancel that note, type a zero (0).

```
0 A=INT(J/4096)
1 REM MEMORY PEEK ROUTINE
2 H$="0123456789ABCDEF"
5 PRINT"   (VALUES IN DECIMAL OR HEX)"
7 DATA 1,16,256,4096
10 GOSUB3500
11 IFLEFT$(LN$,1)<>"H"THENLN=VAL(LN$):GOTO120ELSELN$=RIGHT$(LN$,LE
   N(LN$)-1)
12 FORP1=0TO1
13 L$=MID$(LN$,LE+1-X,1)
14 FORP2=0TO15
15 NEXT:GOTO100
16 Q=P+16*P1+P2
17 NEXTX
18 IFQ>65535 THEN 1090
19 PRINT@P1*16+P2*32,"";:IFHE$="H"THENJ=Q:GOSUB3000ELSEPRINTQ;
20 J=J-4096*A
21 PRINT@P1*16+7+P2*32,"";
22 IFIN$="H"THENHE$="H":RETURN
23 IFIN$="D"THENHE$="D":RETURN
24 IFHE$="H"THENJ=X:GOSUB3050ELSEPRINTX;
25 IFASC(IN$)=12 THENCLS:GOTO110
29 PRINT@P1*16+12+P2*32,"";
30 A=INT(J/256):GOSUB3500
32 PRINT"OR THE '-' KEY TO SCROLL BACK-
34 PRINT"WARDS."
36 PRINT"   PRESS 'H' TO CHANGE TO HEX   OR 'D' TO SWITCH TO DECIM
   AL."
40 J=J-256*A
42 PRINT"   TO CHANGE START ADDRESS,":PRINT"PRESS 'CLEAR'."
50 A=INT(J/16):GOSUB3500
60 A=J-16*A:GOSUB3500
70 RETURN
90 GOSUB2000:CLS
100 IFIN$=";"THENP=P+32:GOTO1150
105 IFIN$="H"ORIN$="D"THEN1150
120 IFIN$="-"THENP=P-32ELSECLS:GOTO110
150 IFP<0 THEN P=0
160 IFP>65535THENP=65525
170 GOTO1012
200 END
500 PRINTMID$(H$,A+1,1);:RETURN
```

Program Listing 1

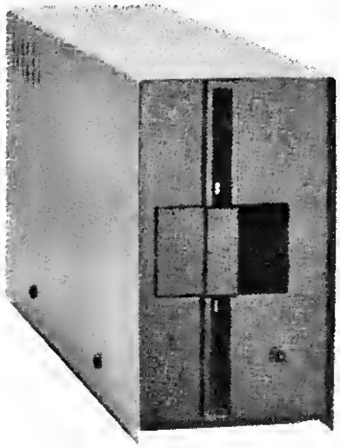
```
              flat   note   sharp
          E    47     48
          D------45-----46-----47------
          C    43     44     45
          B------42-----43-----44------
          A    40     41     42
          G------38-----39-----40------
          F    36     37     38
          E------35-----36-----37------
          D    33     34     35
          C------31-----32-----33------
          B    30     31     32
          A------28-----29-----30------
          G    26     27     28
F----------------24-----25-----26-------------
E                23     24     25
D----------------21-----22-----23-------------
C                19     20     21
B----------------18-----19-----20-------------
A                16     17     18
G----------------14-----15-----16-------------
F                12     13     14
E----------------11-----12-----13-------------
          D     9     10     11
          C-------7------8------9------
          B     6      7      8
          A-------4------5------6------
          G     2      3      4
          F--------------1------2------
```

Fig. 3. Notes for Music-Music-Music.

Mini-Disk Storage Systems for TRS-80* Computers

Access... Offers More. Charges less.

from **$295**



Check this line-by-line comparison of Access 5-inch mini-disk systems for Model I and Model III computers. *Nobody* offers more benefits, better service or lower prices.

TOLL-FREE ORDER NO.
1-800-527-4196†
(orders and literature only)

Company/Drive Model	Price 1/	Flippy	Formatted Diskette Storage Capacity: Dbl-Density	Formatted Diskette Storage Capacity: Sgl-Density	100% Tested	48 hr. Burn-In	Warranty	Trial Period
40-TRACK DRIVES								
Access Unlimited								
AFD-100[1]	$295.00	no	180 Kbytes	102 Kbytes	yes	yes	yes	yes
AFD-100F[1]	329.00	yes	360 Kbytes	204 Kbytes	yes	yes	yes	yes
MTI								
TF-5	359.00	no	?	?	?	?	?	no
Midwest Comp. & Per								
MPI-B-51	321.00	no	?	102 Kbytes	?	?	yes	no
Aerocomp								
Mdl 40-1	349.95	yes	?	?	yes	?	yes	yes
CPU Shop								
CCI-100	314.00	no	?	102 Kbytes	?	?	yes	no
AMI								
40-track	325.00	no	?	?	?	?	?	no
80-TRACK DRIVES								
Access Unlimited								
AFD-200[1]	429.95	no	368 Kbytes	205 Kbytes	yes	yes	yes	yes
AFD-200F[1]	449.95	yes	736 Kbytes	410 Kbytes	yes	yes	yes	yes
MTI								
TF-8	639.00	no	?	200 Kbytes	?	?	?	no
Aerocomp								
80-tk mdl	459.95	yes	?	?	yes	?	yes	yes
CPU Shop								
CCI-280	429.00	no	?	204 Kbytes	?	?	yes	no
AMI								
80-track	560.00	no	?	?	?	?	?	no

1 As advertised in 80 Microcomputing, Jan. 1981.

Free trial offer

Use your brand new AFD drive system for up to 15 days. If you're not *completely* satisfied, pack it in the original shipping container and send it back to Access. We'll refund the price of the system (less shipping charges), no questions asked. (No refund for misuse or improper handling.)

Operating burn-in test — too

Every drive that leaves Access is not only 100% electrically tested and double-checked for mechanical alignment, but it is also given a full 48-hour operating burn-in test. You'll find a test list — checked and signed by one of our competent technicians — in the drive carton. If a drive has latent defects, the burn-in life test will weed them out. The drives we ship just keep on running. And running. And running.

About our warranty

Venus de Milo has about the right number of fingers to count our warranty returns. Nevertheless, your new drive system is covered by our comprehensive 90-day limited warranty. The details are spelled out in the illustrated users manual included with each AFD drive system.

Software galore!

Games. Business programs. System software. We sell many of the best TRS-80* Model I and Model III programs, and at competitive prices. Make Access Unlimited your one-stop shopping center for all of your TRS-80* software, hardware and accessories. Save big! Call our toll-free order number, 1-800-527-4196† for free descriptive literature.

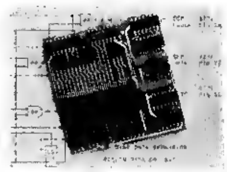
USE YOUR CREDIT CARD AND SAVE! VISA AND MASTER CARD CHARGES ARE NOT DEPOSITED UNTIL THE DAY YOUR ORDER IS SHIPPED. CALL NOW TOLL-FREE, 1-800-527-4196†.

the DOUBLER™

Percom's new plug-in adapter for your Expansion Interface stores almost twice the data on a diskette track as a single-density system. You can store up to four times more data — depending on the type of drive — on one side of a diskette than you can store using a standard Model I mini-disk drive. Other features: Reads, writes and formats *either* single or double density minidiskettes. • Runs TRSDOS*, NEWDOS+‡ Percom OS-80™ or other single-density software *without* changing either software or hardware. Switch to double-density when convenient. • Includes DBLDOS™, a TRSDOS* compatible double-density operating system. • Includes on card, high-performance data separator circuit. • Installs *without* rewiring or trace cutting. • Introductory price, including DBLDOS and format conversion utility, only $219.95.

Permits Model III software to be read on Model I computers.

DATA SEPARATOR™



This PC board plug-in adapter for the TRS-80* virtually eliminates data read errors (CRC error — Track locked out!) which occur on high-density inner disk tracks, a problem that has plagued TRS-80* systems. The Percom Data Separator™ is installed in the Expansion Interface without modifying the host system. Caution: Opening the TRS-80* Expansion Interface may void the limited 90-day warranty: $29.95.

Disk System Interconnecting Cables

Improvement over RS cable design places drive 0, which includes the cable termination, at the end of the cable to eliminate the reflected noise of an unterminated cable. Better data integrity. Prices:

Two-Drive Cable $24.95
Four-Drive Cable 34.95

Power Line Filter

115/250 V, 50-400 Hz. Instructions included for easy installation in standard mini-box chassis: $19.95

Minidiskettes (Double-Density rated)

10 Disks in a convenient plastic organizer box $34.90
Single Disk 3.49

Disk Drive ID Tabs

1" x 1¼" self-adhering plastic drive identification tabs. Compatible silver with engraved black drive number. Two tabs (Nos. 0, 1): $2.50; three tabs (Nos. 0, 1, 2): $3.25, four tabs (0, 1, 2, 3): $4.50.

How to Order Order by calling Access Unlimited toll-free on 1-800-527-4196†. Mail orders also accepted. Orders may be charged to a VISA or MasterCard account or paid by a cashier's check, certified check or money order. We accept COD orders with 25% deposit. Sorry, we cannot accept personal checks. We pay shipping and insurance charges on orders over $1,000.00. Add approximate insurance and shipping charges for under $1,000.00. If in doubt about these charges, ask when you call in your order. Texas residents include 5% sales tax. Minimum order $20.00. Allow 2 to 4 weeks for delivery. †Texas residents call (214) 494-0206.

ACCESS UNLIMITED ✓229

315 N. Shiloh · Ste. D1 · Garland, TX 75042
(214) 494-0206

"Machine language programs load with the CLOADM command rather than SYSTEM."

.. ..	.. .#	.. #.	.. ##	.# ..	.# .#	.# #.	.# ##	#. ..	#. .#	#. #.	#. ##	## ..	## .#	## #.	## ##	COLOR
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	GREEN
80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F	
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	YELLOW
90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F	
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	BLUE
A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF	
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	RED
B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF	
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	WHITE
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF	
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	CYAN
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF	
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	MAGENTA
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF	
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	ORANGE
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF	

Table 6. Low-resolution Graphics Values for CHR$ and POKE (Decimal and Hexadecimal).

Decimal	Hexadecimal	Content
23-24	0017-0018	BASIC stack pointer
25-26	0019-001A	Start of BASIC pointer (normally 1537)
27-28	001B-001C	Start of simple variables pointer
29-30	001D-001E	Start of arrays pointer
31-32	001F-0020	Start of free space pointer
33-34	0021-0022	Bottom of string space pointer
35-36	0023-0024	Pointer to next available byte in string space
37-38	0025-0026	Pointer to start of last string entered
39-40	0027-0028	Top of string space pointer
47-48	002F-0030	Pointer to current line
51-52	0033-0034	Pointer to byte following last data element read
53-54	0035-0036	Pointer to next byte in keyboard buffer
116-117	0074-0075	Pointer to top of available memory (minus one)
135	0087	ASCII value of key last pressed (Returns to zero after INPUT or INKEY$)
136-137	0088-0089	Location of cursor in memory
140	008C	Stores pitch for SOUND command
141-142	008D-008E	Stores pitch times duration for SOUND command
157-158	009D-009E	Used by EXEC command to store jump loction
166-167	00A6-00A7	Pointer to what part of BASIC program is being executed
256	0100	SWI3 routine
259	0103	SWI2 routine
262	0106	SWI routine
265	0109	NMI routine
268	010C	IRQ routine
271	010F	FIRQ routine
275-276	0113-0114	Pointer to start of USR program
338-345	0152-0159	Keyboard map
466-473	01D2-01D9	Stores name specified by CSAVE, CLOAD or SKIPF
474-481	01DA-01E1	Stores name of latest file encountered on tape
733-980	02DD-03D4	Keyboard buffer
1024-1535	0400-05FF	Normal video PEEK/POKE locations
1537	0601	Normal start of BASIC
4095	0FFF	End of 4K user memory
16383	3FFF	End of 16K user memory

Table 7. TRS-80 Color Computer RAM Map

"To whet your appetite a little further, look at... the list of features Radio Shack is claiming...."

A whole note is worth 32, a half note is 16, a quarter note is 8, an eighth note is 4, a sixteenth note is 2 and a thirty-second note is 1. After entering the duration, the note will sound and the orange block will disappear.

Fig. 2 contains a map of your computerized musical keyboard (for a normal G clef). The middle column is the number of the regular note found on that line or in that space. The number to the right is to be used if that note is sharped (#) and the number to the left is used if that note is flat (b). The notes themselves are also indicated. Those of you who do not know music at all will be able to enter songs from a book using Fig. 2 as a graphic quide. In additon, Fig. 3 shows what different notes and rests look like in music books and what numbers should be used for their durations.

When you opt to edit your selection, you will be asked for the starting note number. You will then be given complete instructions on the screen. You will be given the number of the note, the note itself and its duration. You will be able to replay the note (R), change the note (N), alter the duration of the note (D), move ahead one note (right arrow), back up one note (left arrow) or quit editing (Q). (Press the appropriate key only here—do not hit enter.)

When you want to save your composition to tape be sure to depress both play and record buttons on your recorder. Supply the program with the file name to be used. When recalling a program from tape, depress the play button on your recorder and type in the file name of the selection you desire.

Symbol	Name	Duration
	1/32 Note	1
	1/16 Note	2
	1/8 Note	4
	3/16 Note	6
	1/4 Note	8
	3/8 Note	12
	1/2 Note	16
	3/4 Note	24
	Whole Note	32
	Whole Rest	32
	1/2 Rest	16
or	1/4 Rest	8
	1/8 Rest	4
	1/16 Rest	2

Fig. 4. Note Durations

Extended BASIC to Come

Even with only the low level BASIC to work with, the sound and color graphics features of the Color Computer are enormous fun. If you are looking for an inexpensive color computer which can be expanded to do some moderately sophisticated programming, the TRS-80 Color looks like a good buy.

To whet your appetite a little further, look at this list of features Radio Shack is claiming for the extended BASIC:

Five graphics modes up to 192 x 256
Complex sounds with more than one note
Save image from screen
Display predefined image
Zoom image in or out
Move image around screen
Rotate image
Draw line between two points
Draw a circle
Draw a rectangle
Draw a box
Return the time
Print numbers in dollars and cents format
Line editing
Specific error messages
User-definable keys■

No.	CHR$	POKE g/b	No.	CHR$ b/g	POKE g/b	No.	CHR$ b/g	POKE b/g	No.	CHR$ g/b	POKE b/g
0		@	32	spa	spa	64	@	@	96	@	spa
1		A	33	!	!	65	A	A	97	A	!
2		B	34	"	"	66	B	B	96	B	"
3		C	35	#	#	67	C	C	99	C	#
4		D	36	$	$	68	D	D	100	D	$
5		E	37	%	%	69	E	E	101	E	%
6		F	38	&	&	70	F	F	102	F	&
7		G	39	'	'	71	G	G	103	G	'
8	bksp	H	40	(	(	72	H	H	104	H	(
9		I	41	)	)	73	I	I	105	I	)
10		J	42	*	*	74	J	J	106	J	*
11		K	43	+	+	75	K	K	107	K	+
12		L	44	,	,	76	L	L	106	L	,
13	c.ret	M	45	—	—	77	M	M	106	M	—
14		N	46	.	.	78	N	N	110	N	.
15		O	47	/	/	79	O	O	111	O	/
16		P	48	0	0	80	P	P	112	P	0
17		O	49	1	1	81	O	O	113	O	1
18		R	50	2	2	82	R	R	114	R	2
19		S	51	3	3	83	S	S	115	S	3
20		T	52	4	4	84	T	T	116	T	4
21		U	53	5	5	85	U	U	117	U	5
22		V	54	6	6	86	V	V	118	V	6
23		W	55	7	7	87	W	W	119	W	7
24		X	56	8	8	88	X	X	120	X	8
25		Y	57	9	9	89	Y	Y	121	Y	9
26		Z	58	:	:	90	Z	Z	122	Z	:
27		[	59	;	;	91	[	[	123	[	;
28		\	60	<	<	92	\	\	124	\	<
29		]	61	=	=	93	]	]	125	]	=
30		↑	62	>	>	94	↑	↑	126	↑	>
31		←	63	?	?	95	←	←	127	←	?

NOTE: "b/g" means black print on green screen
"g/b" means green print on black screen

Table 8. CHR$ and POKE Values 0 to 127.

Program Listing 2

```
1Ø '   WINDOWS
2Ø '   BOB NICHOLAS
3Ø '   1Ø/Ø9/8Ø
4Ø CLS
1ØØ INPUT"ENTER FORMAT (1-2)";FH$
1Ø5 IFFH$=""THENCLS:GOTO1ØØELSEFM=VAL(FH$)
11Ø IF FH<1 OR FH>2 THEN CLS:GOTO1ØØ
12Ø PRINT
125 INPUT"HOW HANY POINTS (1Ø-1ØØ)";P$
127 IFP$=""THENCLS:GOTO125 ELSEP=VAL(P$)
13Ø IFP<1Ø OR P>1ØØ THEN CLS:GOTO125
135 CLSØ
137 FORX=1TO8
```

Program continues

```
139 PRINT@X*32+2Ø,X;
14Ø FORA=ØTO3Ø:SET(A,X*2,X)
142 NEXTA,X
144 PRINT@384,CHR$(11):PRINT@384,"";
145 INPUT"WHICH COLORS FOR POINTS";C$
146 IFC$=""THEN135
15Ø CLSØ:N=Ø
16Ø ON FM GOTO18Ø,3ØØ
17Ø '  FORMAT ONE - 16 WINDOWS
18Ø X=RND(14)-1:Y=RND(7)-1
19Ø C=VAL(MID$(C$,RND(LEN(C$)),1))
192 IFC<Ø OR C>8 THEN19Ø
195 N=N+1:IFN>P THEN5ØØ
2ØØ FORA=ØTO3
21Ø FORB=ØTO3
22Ø SET(A*16+X,B*8+Y,C)
23Ø NEXTB,A
235 SOUND25Ø-2*N,1
24Ø GOTO18Ø
29Ø '  FORMAT TWO - FOUR KALEIDO       SCOPE WINDOWS
3ØØ X=RND(15):Y=RND(7)
31Ø C=VAL(MID$(C$,RND(LEN(C$)),1))
315 IF C<Ø OR C>8 THEN31Ø
32Ø N=N+1:IFN>P THEN 5ØØ
33Ø FOR A=ØTO1
34Ø FORB=ØTO1
35Ø A1=A*32:B1=B*16
36Ø SET(A1+X,B1+Y,C)
37Ø SET(A1+31-X,B1+Y,C)
38Ø SET(A1+X,B1+15-Y,C)
39Ø SET(A1+31-X,B1+15-Y,C)
4ØØ NEXTB,A
4Ø5 SOUND25Ø-N*2,1
41Ø GOTO3ØØ
5ØØ IN$=""
5Ø5 SOUND15Ø,3Ø
51Ø FORN=1 TO 75Ø
52Ø IN$=INKEY$:IFIN$=""THEN55Ø
53Ø IFASC(IN$)=12 THEN4Ø
55Ø NEXT:GOTO15Ø
```

Program Listing 3

```
4Ø CLS
41 PRINT@7,"MUSIC-MUSIC-MUSIC":PRINT@72,"BY BOB NICHOLAS"
42 NN=75Ø
5Ø DIM N(48),N$(48),P(NN),D(NN)
55 DIM SP(48)
2ØØ FORN=1TO48:READN(N):NEXT
21Ø DATA 5,19,32,45,58,69,78,89,99,1Ø8,117,125,133,14Ø,147,153,159
    ,165,17Ø,176,18Ø,185,189,193
22Ø DATA 197,2ØØ,2Ø4,2Ø7,21Ø,213,216,218,221,223,225,227,229,231,2
    32,234,236,237,238,239,241,242,243,244
23Ø FORN=1TO48:READN$(N):NEXT
235 N$(Ø)="<R>"
24Ø DATA F,F#,G,G#,A,A#,B,C,C#,D,D#,E,F,F#,G,G#,A,A#,B,C,C#,D,D#,E
    ,F,F#,G,G#,A,A#,B,C,C#,D,D#,E,F,F#,G,G#,A,A#,B,C,C#,D,D#,E
25Ø K$="":FORX=1TO27:K$=K$+CHR$(2Ø7):NEXT
26Ø FORN=1TO24:READSP(N):SP(N+24)=SP(N)-16Ø:NEXT
265 DATA 322,259,324,261,326,263,328,33Ø,267,332,269,334,336,273,3
    38,275,34Ø,277,342,344,281,346,283,348
27Ø N=Ø
29Ø '    MENU
3ØØ CLS
3Ø5 PRINT@7,"MUSIC-MUSIC-MUSIC"
31Ø PRINT:PRINT"      1 - ENTER MUSIC
315 PRINT"      2 - CHANGE MUSIC
317 PRINT"      3 - ADD TO MUSIC
32Ø PRINT"      4 - PLAY MUSIC
325 PRINT"      5 - SAVE TO TAPE
33Ø PRINT"      6 - LOAD FROM TAPE
34Ø PRINT@352,"YOUR CHOICE";:INPUTCH$:IFCH$=""THENPRINT@352,CHR$(1
    1):GOTO34Ø
35Ø CH=VAL(CH$)
355 IF CH<1 OR CH>6 THEN PRINT@352,CHR$(11):GOTO34Ø
36Ø ON CH GOTO 5ØØ,1ØØØ,4ØØ,15ØØ,2ØØØ,25ØØ
4ØØ GOSUB15Ø5
41Ø N=N+1:GOTO51Ø
49Ø ' INPUT MUSIC ROUTINE
5ØØ N=Ø:GOSUB15Ø5
51Ø PRINT@416,"'C'=CANCEL NOTE, ^=END SESSION"
52Ø PRINT@448,CHR$(11):PRINT@448,"NOTE # (1-48 OR Ø FOR REST)";:IN
```

Program continues

MICROCOMPUTER TECHNOLOGY INCORPORATED

SPRING SALE

CLEARANCE SPECIAL

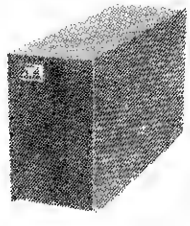
SHUGART SA400 (TF-3)

35 track disk drive includes power supply and chassis.
LIMITED SUPPLY

$299

FREE FULL YEAR WARRANTY

with purchase of new SHUGART SA400 (TF-3L) 40TK comes complete with power supply, chassis cable & NEWDOS 80
ONLY **$439**



THE MICROCONDUCTOR™

the ultimate data base manager for your TRS-80®

The MICROCONDUCTOR™ is not just a file manager but a true Data Base Management System suitable for both the novice and professional users.

DATA FILES—No limit on the number of records a file can have
FIELDS—Any type (string interger single double) Eight entry modes (including defaults, counting, and suppress)
REPORTS—Four ways to generate reports Total numeric column(s). Print on any paper in any format (statements labels etc)
SORT—Any field(s) (i.e. multiple-key sort) Any size file numeric or ASCII Ascending or descending
MAINT.—Command anticipation Record duplication. Direct access and sequential search
UPDATING/MERGING—Add subtract multiply, divide fields Combine results from previous calculations Test for any condition and take action

TRS-80® MODEL I	$159
TRS-80® MODEL II	$399

NEW FOR TRS-80®

TF-8 80 TRACK DISK DRIVE by MPI
Double your capacity. Single head mini floppy. More than 200K bytes of storage. Complete with power supply and chassis.
$619

TF-9 DUAL 80 TRACK DISK DRIVE by MPI
Quadruple your capacity. 400K bytes of storage (like having 4-40 track drives in one unit). Complete with power supply and chassis.
$769

MOD II 8" Disk System

- 1 Drive System $949
- 2 Drive Expansion System $1,445

Disk Drive Sale!

with FREE power supply and chassis.

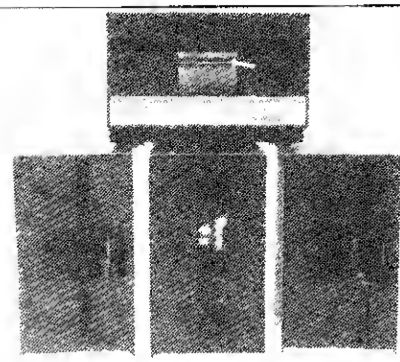
Shugart SA400L, 40 trk.	$329
Pertec FD200, 40 trk.	$379
TF-5 MPI B51, 40 trk.	$349
TF-7 Mircopolis, 77 trk.	$559
TDH-1 Dual sided drive, 35 trk.	$449
TF-3M Drive sys. 2 Shugart 40 trk.	$649

SOFTWARE SALE

NEWDOS 80	$149
NEWDOS+ 40 trk.	$110
30 trk.	$99
AJA Business Pkg.	$359
The Source	$100
Basic Compiler	$195

MONTHLY SPECIAL

Shugart SA801 Bare Drive **$459**



MOD I Disk Expansion Systems

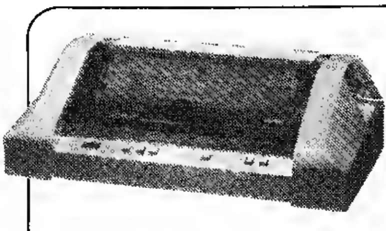
- 2 Shugart SA400 TF-3L $658
- 1 Two-Drive Cable $26
- 1 Expansion Interface 32K $439
- 1 DOS+ $99

TOTAL LIST PRICE $1,222
SPECIAL PRICE ONLY **$1,077**

★BARE DRIVES FOR ANY MICROCOMPUTER

Pertec FD200	$282	FD250	$359
Shugart SA400 35 track	$269	SA400L 40 track	$279
MPI B51	$279	B52	$349
MPI B91	$399	Tandon 40TK.	$262

Memory Kit (16K)	$43.00
AC Isolator (6 socket)	$49.95
Disk Head Cleaner	$19.95
Diskettes Verbatim (10) hard ring	$30.00



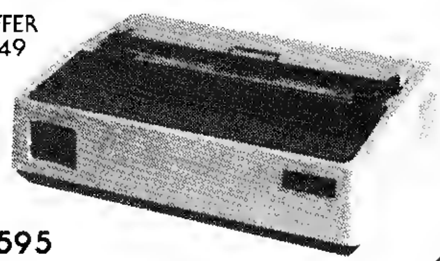
Anadex 9500 $1,449

Printers

Microline 82	$849
Microline 83	$1,060
Centronics 779	$1,069
Centronics 700	$1,129
Base 2	$649
Centronics 737	$839
Centronics 702-9	$1,995
Malibu	$2,093
Paper Tiger	$895
Daisy Wheel	$1,799

INTRODUCTORY OFFER
SAVE $300 LIST $949

Okidata Microline 80
$599



EPSON PRINTER $595

® Registered trademark of Radio Shack

```
     PUTP$
525 IFP$=""THEN520
526 IF P$="C"THENN=N-1:GOTO520
527 IFP$="^"THENN=N-1:GOTO300
528 P(N)=VAL(P$)
530 IF P(N)<0 OR P(N)>48 THEN520
535 IFP(N)=0 THEN545
540 PRINT@SP(P(N)),CHR$(255);
545 PRINT@416,CHR$(11):PRINT@416,"0 TO CANCEL THIS NOTE";
550 PRINT@448,CHR$(11):PRINT@448,"DURATION # (1-32)";:INPUTD(N)
560 IFD(N)<0 OR D(N)>32 THEN550
561 IFP(N)=0 THEN566
562 IFD(N)>0 THENSOUNDN(P(N)),D(N)
564 IF LEN(N$(P(N)))>1 THENPRINT@SP(P(N)),CHR$(128);ELSEPRINT@SP(P
     (N)),CHR$(207);
566 IFD(N)=0 THEN510
570 IF N>NN THEN CLS:PRINT"MAXIMUM # OF NOTES ENTERED.":GOSUB5000:
     GOTO300
580 N=N+1:GOTO510
990 '    CHANGE MUSIC
1000 CLS:PRINT"START WITH WHICH NOTE             (0 -"N")";:INPUTZ5
1002 IF Z5<0 OR Z5>N THEN1000
1005 FORX=Z5 TO N
1010 IFP(X)>0 AND D(X)>0 THEN SOUNDN(P(X)),D(X)
1015 CLS
1016 PRINT"NOTE #"X
1017 PRINT"NOTE ="P(X)"  DURATION ="D(X)
1018 PRINT
1020 PRINT"R = REPLAY NOTE                   N = CHANGE NOTE
           D = CHANGE DURATION                = BACK UP ONE NOTE
1022 PRINT"RIGHT ARROW = AHEAD ONE NOTE
1025 PRINT"'Q' = QUIT EDITING
1027 PRINT:PRINT"PRESS KEY OF YOUR CHOICE."
1030 F$=""
1040 F$=INKEY$:IFF$=""THEN1040
1050 IFF$="R"THEN1010
1052 IFASC(F$)<>8 THEN1060
1054 X=X-1:IFX<0 THENX=0
1056 GOTO1010
1060 IFASC(F$)=9 THEN1200
1070 IFF$="N"THEN1150
1075 IFF$="Q"THEN300
1080 IFF$<>"D"THEN1015
1085 PRINT:INPUT"NEW DURATION # (1-32)";V
1090 IFV<1ORV>32THEN1000
1100 D(X)=V:GOTO1010
1150 PRINT:INPUT"ENTER NEW NOTE # (1-48)          (0 FOR REST)";V
1155 IF V<0 OR V>48 THEN1000
1160 P(X)=V:GOTO1010
1200 NEXT
1210 GOTO300
1490 '  PLAY THE MUSIC
1500 GOSUB1505:GOTO1600
1505 CLS0
1510 PRINT@7,"MUSIC-MUSIC-MUSIC";
1520 FORA=0TO1:FORB=0TO3
1525 PRINT@A*160+66+B*32,K$;
1530 NEXTB
1535 PRINT@A*160+194,"F G A B C D E F G A B C D E";
1540 NEXTA
1550 FORX=1TO48:IFLEN(N$(X))>1THENPRINT@SP(X),CHR$(128);:PRINT@SP(
     X)-32,CHR$(128);
1555 NEXT:RETURN
1600 FORX=0 TO N
1610 IF P(X)=0 THEN FORTI=1TOD(X)*5:NEXT:GOTO1700
1615 PRINT@SP(P(X)),CHR$(255);
1620 SOUNDN(P(X)),D(X)
1625 IF LEN(N$(P(X)))>1 THENPRINT@SP(P(X)),CHR$(128);ELSEPRINT@SP(
     P(X)),CHR$(207);
1700 NEXTX
1710 PRINT@483,"'R' = REPLAY, 'M' = MENU ";
1720 IN$=""
1730 IN$=INKEY$:IFIN$=""THEN1730
1740 IF IN$="R"THEN1600
1750 IFIN$="M"THEN300
1760 GOTO1730
1990 '   SAVE TO TAPE
2000 CLS
2005 GOSUB3000:CLS
2010 PRINT@230,"SAVING "FS$" TO TAPE."
2020 OPEN"O",-1,FS$
2030 PRINT#-1,N;
2040 FORX=0TON
2050 PRINT#-1,P(X);D(X)
2060 NEXTX
2065 CLOSE
2070 CLS:PRINT@230,"MUSIC IS SAVED!":GOSUB5000:GOTO300
2490 '   LOAD FROM TAPE
2500 CLS
```

Program continues

```
2505 GOSUB3000:CLS
2507 PRINT@230,"LOADING "FS$"."
2510 OPEN"I",-1,FS$
2520 INPUT#-1,N
2530 FORX=0TON
2540 INPUT#-1,P(X),D(X)
2550 NEXTX
2555 CLOSE
2560 CLS:PRINT@230,"MUSIC IS LOADED!":GOSUB5000:GOTO300
2990 '    ENTER FILESPEC
3000 INPUT"ENTER FILESPEC";FS$
3010 IFFS$=""THEN3000
3020 IF LEN(FS$)>8 THEN 3000
3030 RETURN
4990 ' PAUSE ROUTINE
5000 IN$=""
5010 PRINT@482,"PRESS <CLEAR> TO CONTINUE.";
5020 IN$=INKEY$:IFIN$=""THEN5020
5030 IFASC(IN$)=12 THEN RETURN ELSE5020
```

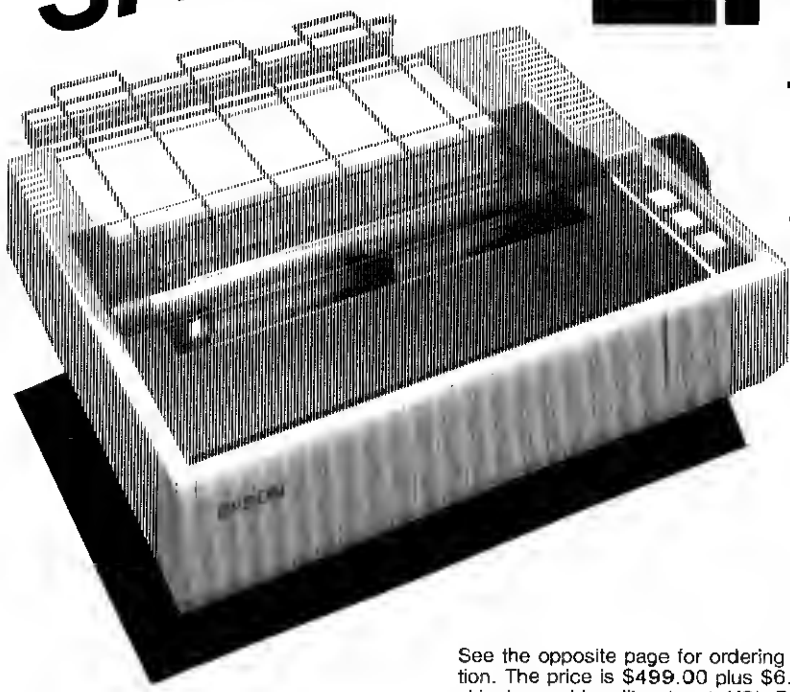

✓See List of Advertisers on page 306

SAVE!

EPSON

MX-80

$499.00



See the opposite page for ordering information. The price is $499.00 plus $6.00 for shipping and handling (cont. US). Does not include cable. Complete with David Lien manual.

THE REAL THING

Yes, this is the EPSON MX-80 printer you have read so much about. . . the one everyone's buying. Better get yours while they are available at this terrific price! One of the most advanced 80 column dot matrix printers available featuring the world's first disposable print head. There are lots of printers to choose from but, once you have compared, you will find it hard to pass up the Epson MX-80.

FEATURES

- Bidirectional Printing.
- Logicial seeking of shortest lines.
- 80 Characters per second.
- 64 Block Graphics Characters.
- Forms Handling.
- Disposable print head.
- Adjustable Tractor-type pin feed.
- Paper width 4 to 10 inches.
- Inked ribbon cartridge with 3 million character life.
- Self test mode.
- Full 96 character ASCII with decenders.
- Printing Modes: Standard, Double, Emphasized and Double Emphasized.
- Correspondence Quality.

EXTRAS

TRS80 Cable	$30.00
Apple Cable& Interface	$96.00
Ribbon Cartridge, Black	$13.95
RS 232 Interface	$69.95

ORDER TODAY!!!

ANOTHER CHOICE-OKIDATA MICROLINE 80

Here is another fine quality printer available at the outstanding price of $499.00. The Microline 80 has many outstanding features:

- Unidirectional Printing.
- 80 Characters per second.
- 64 Block Graphics Characters.
- Forms Handling.
- Tractor, Pin and Friction Feed.
- Paper Width 4 to 9 inches.
- Self Test Mode.
- Full 96 Character ASCII (no decenders).
- Printing Modes: Standard, Double and Condensed.

Add $6.00 shipping. Does not include cable.

CABLES, SOFTWARE & OTHER GOODIES

- **DRIVE CABLES**

2-Drive	$26.95
4-Drive	$36.95
Extender	$16.95

- **DISKETTES** (5-1/4 In, soft sectored. box of 10)

Cert. one side, single density	$29.95
Cert. two sides, double density	$39.95

- **OPERATING SYSTEMS**

DOSPLUS 3.2, 3.2D	$99.95
NEWDOS/80	$149.00
TRSDOS 3.2 DISK & MANUAL	$17.95
Double Zap II (spec. Newdos/80 or VTOS)	$42.50
VTOS 4.0 MANUAL	$25.00

- **OTHER SOFTWARE & BOOKS**

SCRIPSIT (disk)	$89.95
SPECIAL DELIVERY	$119.95
XTRA SPECIAL DELIVERY	$189.95
SUPER UTILITY	$39.95
MICROSOFT BASIC DECODED & OTHER MYSTERIES	$25.95
TRS80 DISK & OTHER MYSTERIES	$19.95
MAKE 80	$14.95

SPECIAL

* * * * * * * * * *

WORD PROCESSING PACKAGE

Includes the following:

EPSON MX-80 PRINTER & CABLE SCRIPTSIT (disk), SPECIAL DELIVERY (disk) & Freight and Insurance.

Special

$725.00

(Reg. $917.00)

ORDER TODAY FROM

AEROCOMP

Redbird Airport, Bldg. 8
P.O. Box 24B29
Dallas, TX 75224 ✓387

See Opposite Page ⇨

40 & 80 Track "FLIPPY" Drives for TRS-80*

REAL VALUE

AEROCOMP offers the best value in microcomputer disc drives on the market today! Reliability, features and cost tough to beat. We deliver...and we stand behind our products, as evidenced by the only FREE TRIAL OFFER in the industry. Examine your systems needs and order today!

MYSTERY REMOVED

There appears to be some confusion in the terminology used to describe disc drives and their features. Here's what we mean:

- **FLIPPY** Allows the use of both sides of a diskette with a single-headed drive by simply turning the diskette over (model 40-1&80-1).
- **TRACK DENSITY** Specified in tracks per inch (TPI). Refers to the number of tracks per radial inch on the diskette. Typically 48 TPI=40 usable tracks and 96 TPI=80 usable tracks.
- **DOUBLE DENSITY** Refers to recording density in bits per inch (bpi). Typically single density means data can be recorded up to 2,938 bpi; double density means data can be recourded up to 5,876 bpi.
- **DOUBLE SIDED** Refers to number of read/write heads. Single-sided is one head, read/write one side only; double-sided is dual heads allowing read/write operations on both sides of the diskette. A double sided drive appears as two seperate drives to the controller. (Model 80-2 & 160-2)
- **ACCESS TIME** The time required for the head to move from one track to the next. Typacilly 5 to 40 milliseconds (ms).

SELECT YOURS!

- **40-Track "FLIPPY" $349.95** (Model 40-1) 48 TPI, single dens. 125K bytes/side, double dens. 250K/side.
- **80-TRACK "FLIPPY" $459.95** (Mod. 80-1) 96 TPI, single dens. 250K bytes/side, double dens. 500K/side.
- **40 TRACK DUAL HEAD $459.95** (Mod. 80-2) 48 TPI, double-sided, 40 tracks/side; single dens. 250K bytes, double dens. 500K.
- **80 TRACK DUAL HEAD $599.95** (Mod. 160-2) 96 TPI, double-sided, 80 tracks/side; single dens. 500K bytes, double dens. 1 megabytes.

NOTE: All capacity values are unformated. All models capable of single or double density. All with power supply/silver enclosure. 115 VAC, 50-60 Hz. 115/230 VAC available.

STARTER A

40-Track drive, 2-drive cable, TRSDOS 2.3 Disk & Manual, Freight & Ins.

$339.95

(Reg. $380.00)

STARTER B

40-Track FLIPPY drive, 2-drive cable,TRSDOS 2.3 Disk & Manual,Freight & Ins.

$369.95

(Reg. $401.00)

SPECIAL PACKAGES

#1 40-Track FLIPPY drive
2-Drive cable
Newdos/80
Freight & Ins.$458.00(reg.$528.00)

#2 80-Track FLIPPY drive
2-Drive cable
Newdos/80
Freight & Ins. $569.00 (reg.$638.00)

#3 TWO (2) 40-Track FLIPPY drives
4-Drive cable
Newdos/80
Freight & Ins. $785.00 (REG. $893.00)

#4 TWO (2) 80-Track FLIPPY drives
4-Drive cable
Newdos/80
Freight & Ins. $899.00 (reg.$1113.85)

#5 TWO (2) 40-Track DUAL HEAD Drives
Cable
Newdos/80
Freight & Ins. $995.00 (Reg. $1105.00)

#6 TWO (2) 80-Track DUAL HEAD Drives
Cable
Newdos/80
Freight & Ins. $1,245.00
(Reg. $1,385.00)

FREE TRIAL OFFER

Order your AEROCOMP Disc Drive and use it with your system for up to 14 days. If you are not satisfied for ANY REASON (except misuse or improper handling),return it, packed in the original shipping container, for a full refund.(Special packages will be prorated). We have complete confidence in our products and we know you will be satisfied! **ORDER TODAY!!**

WARRANTY

We offer you a 120 day unconditional warrenty on parts and labor against any defect in materials and workmanship. In the event service, for any reason, becomes nescessary, our service department is fast, friendly and cooperative.

100% TESTED

AEROCOMP Disc Drives are completely assembled at the factory and ready to plug in when you receive them. Each drive is 100% bench tested prior to shipment. We even enclose a copy of the test checklist, signed by the test technician, with every drive. AEROCOMP MEANS RELIABILITY!

ORDER NOW!!

To order by mail, specify Model Number(s) of Drive, cable, etc. (above), enclose check, money order, VISA or MASTERCHARGE card number and expiration date, or request C.O.D. shipment. Texas residents add 5% sales tax. Add $5.00 per drive for shipping and handling. Please allow 2 weeks for personel checks to clear our bank. No personel checks will be accepted on C.O.D. shipments-cash, money orders or certified checks only. You will receive a card showing the exact C.O.D. amount before your shipment arrives. Be sure to include your name and shipping address. WE SHIP PROMPTLY! In the event there is a slight delay, you will be notified of the shipping date and we will NOT charge your bankcard until the day we ship!

COMPARE AND BUY AEROCOMP!

	"FLIPPY"	ACCESS TIME (track to track)	HEAD LOAD SOLENOID	DISC EJECTOR	CAPACITY (unformated single density)	EASY-ENTRY DOOR	FREE TRIAL
AEROCOMP	**YES**	**5ms.**	**YES**	**YES**	**250K bytes (both sides)**	**YES**	**YES**
RADIO SHACK*	NO	40ms.	YES	NO	109K bytes	NO	NO
PERCOM	YES	25ms.	YES	NO	250K bytes (both sides)	YES	NO
MPI	NO	5ms.	YES	YES	125K bytes	YES	NO
SHUGART	NO	40ms.	YES	NO	109K bytes	NO	NO
TANDON	NO	5ms.	NO	NO	125K bytes	NO	NO

Factual material from current manufacturer's data sheets is believed reliable but cannot be guaranteed, comparing Aerocomp Model 40-1 to similar models.

The TRS-80* expansion interface limits the track to track access time to 12ms.

*Trademark of Tandy/Radio Shack.

WRITE AEROCOMP TODAY FOR MORE VALUES !!!

CALL TOLL FREE FOR FAST SERVICE
(800) 824-7888, OPERATOR 24
FOR VISA/MASTERCHARGE/C.O.D. ORDERS
California dial (800) 852-7777, Operator 24. Alaska and Hawaii dial (800) 824-7919, Operator 24.
TOLL FREE LINES WILL ACCEPT ORDERS ONLY!
For Applications and Technical information, call (214) 337-4346 or drop us a card.

Dealers inqiries invited

AEROCOMP
Redbird Airport, Bldg. 8
P.O. Box 24829
Dollas, TX 75224

✓387

✓See List of Advertisers on page 306

NEW!! THE ULTIMATE TRS-80 EXPANSION INTERFACE

CMD Expansion Interface

COMPARE FEATURES QUALITY DEPENDABILITY AVAILABILITY

Complete CMD Kit (includes all * below)	299.00
Complete CMD Kit with Dual Switching Power Supply	359.00

NOTE: The Dual Switching Power Supply is capable of handling 2 disk drives and the expansion interface.

Compare these features to the others!

1. The board is silk screened and solder mashed!
2. A **complete** bound instruction and test manual comes with each kit!
3. Dip switches are used. **No soldered jumpers!**
4. Dip packaged resistors eliminate messy vertically mounted termination resistors!
5. Complete service and support after the sale!

You can purchase either a complete kit or buy the separate components and add on as your budget ALLOWS.

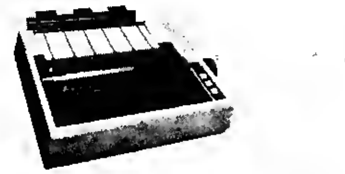
* on board Data Separator	18.55	Port	8.95
* on board Direct Connect Modem	39.95	* 32K Ram Expansion (less Ram)	11.95
* 2K-4K Eprom relocatable/ programmable	17.95	* Cassette Port	3.95
* RS232 Serial Port	17.95	* Analog Power Supply	29.95
* Real Time clock	17.95	Dual Switching Power Supply	74.95
* Floppy Disk Controller	31.95	* CMD Printed Circuit Board	79.95
* Centronics Parallel Line Printer		* Hardware and Socket Kit	19.95
		* Complete Instruction & Test Manual	

Where Can You Get?

1. Biderectional printing with a logic seeking head.
2. 80 CPS capability in a user defined choice of 40, 80, 66, or 132 columns.
3. 96ASC11 with descenders, 64 graphic characters and eight international characters.
4. Correspondence quality printing.

5. Double, Emphasized, and Double Emphasized printing modes.
6. Compressed, Compressed Expanded, Normal Expanded, and Normal print sizes.
7. Paper handling range from 4" to 10."
8. Dependability and fast service when you need it.
9. **AFFORDABILITY.**

OF COURSE. IT IS THE **EPSON MX80**



CALL US TODAY FOR OUR "AFFORDABLE" PRICE

Model III Disk Drive Kits & Drives

Now available from **Texas Digital Inc.**, a Radio Shack Disk Drive Kit for your model III for **only** $429.95.

Compare these standard features of our list to the others.

1. Drive Controller Board
2. All necessary interconnecting cables
3. Switching Power Supply
4. Disk Drive Mounting Brackets
5. Internal Disk Drive Cable
6. TRS DOS Manual
7. Comprehensive installation & theory manual
8. Radio Shack and Tandon TM100 drive compatible
9. 16 K RAM

COMPLETE MODEL III DRIVE PACKAGES

These packages all include, the Model III Disk Drive Kit and the Micro Systems Software Model III DOS.

M3DK-1	(2-TM100-1 Drives) provides over 350K Bytes storage	919.95
M3DK-2	(2-TM100-2 double headed drives) provides over 700K Bytes storage	1,149.95
M3OK-3	(2-TM100-3 80 track drives) provides over 700K Bytes storage	1,199.95
M3DK-4	(2-TM100-4 double headed 80 track drives) provides over 1.4 M Bytes storage	1,399.95

LNW COLOR COMPUTER KIT
$529.95
(less P.C.B.)

Includes all components to assemble the LNW color computer; high resolution video RAM and 16K user RAM, keyboard, and Level II vompatible ROMS. Printed circuit board not included.

LNW EXPANSION INTERFACE KIT
$239.95
(less P.C.B.)

Includes all components to assemble the LNW Expansion Interface plus 32 K of RAM. It does not include the printed circuit board, dual casette relay or cabinet.

Beautiful custom all wood cabinet is available for housing the LNW Expansion Interface.
$99.95

DISK DRIVES - Compare !!!!

RELIABILITY / FEATURES / WARRANTY
PRICE / DELIVERY
Texas Digital offers the best of all with TANDON Disk Drives

Warranty:	120 day repair / replace warranty
Reliability:	8000 hrs. mean time between failure
Features:	5 m.s. Track to Track Access Time
	0 sec. Head load time (no solenoid either!)
	Single or Double Density Capabilities
	40 / 80 Track - Single & Double Sided
	Maintenance Manual Provided with drive

TM100-1 Single headed 40 track disk drive, unformatted double density storage = 250 KB
TRS80tm Storage single density = 102 KB Double = 184KB
Bare drive $225.00 **With case/supply $299.95**

TM100-2 Double headed 40 track drive, unformatted double density storage = 500 KB
TRS80tm Storage single density = 204KB, Double = 367KB
Bare drive $345.00 **With case/supply $419.95**

TM100-3 Single headed 80 track drive, unformatted double density storage 500KB
TRS80tm Storage single density = 204KB, Double = 367KB
Bare drive $375.00 **With case/supply $449.00**

TM100-4 Double headed 80 track drive, unformatted double density storage = 1.0mB
TRS80tm Storage single density = 408KB, Double = 735KB
Bare drive $475.00 **With case/supply $549.00**

2 Drive Disk Cable $24.95
4 Drive Disk Cable $34.95
Drive Extender Cable $13.95

MEMORY/SPECIAL FUNCTION

FD1771B-01	23.95	FD1793B-01	44 95
TR1602	3.95	AY5-1013A	4 95
MC1372	6 46	MC6674	13.00
2716-450	12 95	2114-450	5 95
Z80A	14 95	2102	1 75
CC74S188	Programmed for LNW Color CPU		9.95
CC2716-6	Level II Roms for Color CPU		99 95

LINEAR

LM1488	1.20	TL084	1 75
LM1489	1 20	MC1458	58
UA7805	1 10	MLM311P1	.90
UA7612	1 10	MC14412	17.00
UA7912	1 10	78H05	6.00
79L12	1 50	75452	.35

DIODES TRANSISTORS / SEMICONDUCTORS

1N914/1N4148	.05	KBP02	79
1N4001	09	2N3904	.25
1N5231	.25	2N3906	.25
1N5244	25	MPU131	.89
1N5244	25	J175	.89

CRYSTALS

1.0 MHZ	4 00	3 579 MHZ	3.95
4 0 MHZ	4.00	16.0 MHZ	4 95

1% PRECISION RESISTORS

267 OHMS	324 OHMS	619 OHMS
845 OHMS	1.24 K OHMS	2.55 K OHMS
8 66 K OHMS	8.87 K OHMS	18.7 K OHMS
21.5 K OHMS	95.3 K OHMS	97.6 K OHMS
165 K OHMS	210 K OHMS	237 K OHMS

All 1% Resistors are $ .18 each

I.C. SOCKETS

8 PIN Low Profile	12	20 PIN Low Profile	.23
14 PIN Low Profile	14	22 PIN Low Profile	27
16 PIN Low Profile	16	24 PIN Low Profile	30
18 PIN Low Profile	.21	40 PIN Low Profile	.47

7400 SERIES TTL

7438	.20	74157	.60
7442	.45	74161	.83
7451	.56	74164	.83
7474	.32	74165	.83
7486	.32	74166	1.15
7490	.42	74173	1.15
7492	.42	74175	.75
7493	.42	74176	.75
7495	.58	74194	.83
74121	.32	74195	.64
74123	.45	74151	.54
74125	.45	74366	.64
74132	.41	74367	.66

74LS00 SERIES - LOW POWER SCHOTTKY

74LS00	.27	74LS139	.75
74LS02	.27	74LS153	.75
74LS04	.29	74LS155	.90
74LS05	.39	74LS157	.75
74LS08	27	74LS161	1.10
74LS09	.32	74LS164	1.18
74LS10	.27	74LS166	1.85
74LS11	.71	74LS174	1.89
74LS13	.56	4LS175	.95
74LS14	1.15	74LS193	1.10
74LS15	.31	74LS240	1 65
74LS20	.27	74LS241	1.65
74LS21	.31	74LS244	1.65
74LS30	.26	74LS245	2.25
74LS32	.31	74LS257	.85
74LS74	.44	74LS367	69
74LS86	.41	74LS373	1.89
74LS93	.70	74LS374	1.65
74LS123	1.18	74LS383	2.35
74LS124	1 69	74LS138	.86
74LS132	.95		

74S00 SCHOTTKY

74S04	.55	74S05	47
74S22	.47	74S32	.47
74S64	.47	74S74	.61
74S112	.71	74S161	1.75
74S174	1.59	74S175	1.50
74S280	2.45	74S387	5.86

ONE YEAR GUARANTEE
on our NEC-Motorola 200 N.S 16K Memory Kits.
16K - $23.95
32K - $45.95
48K - $74 95

These are top quality new memory chips at unbelievably low prices.

1. Radio Shack, TRS-80, TRS/DOS are registered trademarks of Tandy Corp.
2. Prices and Specifications are subject to change without notice.

TEXAS DIGITAL INC. ~392

(FORMERLY COMPUTEX)
15502 HIGHWAY 3, SUITE 715
WEBSTER, TEXAS 77598

Visa and Mastercard accepted
Personal and company checks require two weeks to clear

Big things are happening at Computex. (Even our name has changed to Texas Digital, Inc.) We have expanded our facilities and added personnel to better service our customers.

Our service is BIGGER AND BETTER
Our product lines are BIGGER AND BETTER
Our prices....ARE STILL THE LOWEST!
CALL TODAY!!!

(713) 488-8022

Conduct your own on those overloaded disks.

Purge

Joe Ligori
2660 W. Ball Rd. #68
Anaheim, CA 92804

Have you ever sat at your TRS-80 typing KILL FILENAME over and over to clear a diskette of unwanted files? If you've wanted a faster way of clearing a diskette, Purge Utility will fill the bill.

Purge Utility will operate with either TRSDOS or NEWDOS operating systems with only one difference. Under TRSDOS the Purge Utility will not delete a password protected file. Since NEWDOS ignores password protection, Purge Utility will delete them.

Program Operation

Purge Utility is presented here in two forms, an assembly language listing and a BASIC program which POKEs the decimal values into memory and allows the program to be dumped onto disk. It operates simply.

Initially the program prompts the operator for a drive number (0–3) to indicate which diskette is to be purged. This beginning drive number prompt also allows loading the Purge Utility from one diskette, then loading a second diskette that you wish to purge. You can purge any disk without having to actually store the utility on all your disks (useful for single drive systems).

Once the drive prompt is answered, the program displays each file in the directory (system files excluded), and prompts for a Y or N to indicate if the file is to be deleted or not. All your entries are single keystrokes. You don't need to press Enter after your responses. The program can be aborted at any point by hitting an X instead of a Y or N.

The files are displayed in order within the directory; invisible attribute files are also displayed. When all the files have been displayed, the screen clears and the program displays each file as it deletes it, then returns to DOS Ready.

Program Composition

The program uses two DOS routines, Open and Kill. Although BASIC requires a file to be closed before it can be killed, in actuality the file is opened and then deleted from the directory. Three ROM routines are also utilized in the program. CRT (033AH) displays a single character (passed in the accumulator) on the screen and handles all screen positioning. CLS (01C9H) is the clear screen routine in ROM. KBD (0049H) is a keyboard scan routine. This routine scans the keyboard, waits for an entry and returns with the character in the accumulator. The program aborts itself if a drive-not-ready condition occurs, or if an error is encountered when reading the directory.

An internal-sector read routine is used to read the directory rather than a DOS based directory read routine. This helps ensure compatibility on future DOS releases and allows a drive-not-ready check routine.

The directory is located on track 11H (17 decimal) and the actual directory entries reside in sectors two through nine of the directory track. Sectors zero and one contain a granule allocation table and a hash index table. These two sectors are not accessed by the program, but are accessed and updated by the DOS Open and Kill routines called by the program.

The program resides at memory locations 7400H to 7666H. The location was chosen to make the program compatible with the DUMP command when the program is entered with the BASIC program. Once the BASIC program is executed, enter CMD"S" to return to DOS Ready. Enter the following DUMP command:

```
DUMP PURGE/CMD (START=X'7400',
END=X'7666',TRA=X'7400')
```

This will store the program on disk and allow it to be executed

by simply entering Purge from DOS Ready. If you are using an editor/assembler, be sure to use a file name with /CMD as an extension to achieve immediate execution from DOS.

Although this program's usefulness is somewhat limited, I hope that the assembly language listing will be of some educational value to beginning programmers. ■

Program Listing 1 in BASIC

```
1 ' **  BASIC PROGRAM TO POKE PURGE PROGRAM INTO MEMORY
        **
2 ' **  RETURN TO 'DOS READY' AND DUMP THE PROGRAM ONTO
        **
3 ' **  DISK.  THE COMMENTS WITHIN THIS PROGRAM SHOULD
     BE **
4 ' **  DELETED TO ALLOW THE PROGRAM TO FIT IN MEMORY.
        **
5 '
9 '     ** SET MEMORY SIZE FROM WITHIN BASIC PROGRAM **
10 POKE16561,250:POKE16562,115:CLEAR10:CLS
11 '     ** SET START OF PROGRAM (7400H) AND ZERO CHECKS
     UM **
12 S=&H7400:C=0
28 '    ** LOOP TO READ A BYTE (DECIMAL) AND POKE TO MEM
     ORY **
30      FOR I = S  TO  &H7666
40      READ X  :  POKE I , X
45      PRINT@256,"STORING ";X;" INTO MEMORY LOCATION ";
     I;CHR$(30)
47      C = C + X      'ADD UP CHECKSUM
50      NEXT
59 '    ** CHECK THE CHECKSUM **
60 IF C <> 57478 THEN PRINT "CHECKSUM ERROR" : END
70 PRINT "RETURN TO DOS" : PRINT
80 PRINT"DUMP PURGE/CMD (START=X'7400',END=X'7666',TRA=
     X'7400')"
7000 DATA 243,205,201,1,33,191,117,205,201,116,33,84,11
     8,205
7010 DATA 201,116,205,73,0,254,52,48,249,254,48,56,245,
     205
7020 DATA 58,3,214,48,60,50,103,118,205,98,117,205,213,
     116,6
7030 DATA 64,33,0,100,205,62,117,197,229,203,118,32,37,
     203,102
7040 DATA 40,33,17,5,0,25,205,22,117,33,236,117,205,201
     ,116,205
7050 DATA 244,116,254,89,245,205,58,3,205,62,117,241,22
     5,32,3
7060 DATA 62,255,119,229,225,17,32,0,25,193,16,205,205,
     201,1,6
7070 DATA 64,33,0,100,197,229,62,255,190,32,71,205,60,1
     17,17
7080 DATA 5,0,25,17,0,99,6,8,126,254,32,40,2,18,19,35,1
     6,246
7090 DATA 62,47,18,19,6,3,126,254,32,40,5,18,35,19,16,2
     46,62
7100 DATA 13,18,33,253,117,205,201,116,225,229,17,5,0,2
     5,205
7110 DATA 22,117,205,62,117,33,0,126,17,0,99,205,36,68,
     205,44
7120 DATA 68,225,17,32,0,25,193,16,170,205,201,1,195,45
     ,64
7130 DATA 126,254,0,200,229,205,58,3,225,35,24,244,62,1
     7,50
7140 DATA 104,118,62,2,50,105,118,1,0,100,205,110,117,2
     21
7150 DATA 33,105,118,221,52,0,62,10,221,190,0,32,239,20
     1,205,73
7160 DATA 0,254,89,200,254,78,200,254,88,202,195,116,24
     ,240,126
7170 DATA 254,32,40,3,205,16,117,35,16,245,201,221,119,
     0,221,35
7180 DATA 201,6,12,221,33,223,117,205,74,117,221,33,223
     ,117
7190 DATA 6,8,205,4,117,62,32,190,40,10,62,47,205,16,11
     7
7200 DATA 6,3,205,4,117,33,223,117,205,201,116,201,62,1
     3
7210 DATA 205,58,3,201,6,32,221,33,0,99,221,54,0,32,221
7220 DATA 35,16,248,201,197,58,103,118,71,62,128,7,16,2
     53
7230 DATA 50,225,55,193,201,205,83,117,33,0,0,43,124,18
     1
7240 DATA 32,251,201,205,83,117,58,104,118,50,239,55,58
     ,105
7250 DATA 118,50,238,55,33,236,55,62,30,119,197,193,197
     ,193
7260 DATA 126,7,48,3,195,186,113,15,15,56,245,54,136,17
     ,239
7270 DATA 55,197,193,197,193,24,3,15,48,10,126,203,79,4
     0
```

Program continues

```
7280 DATA 248,26,2,3,24,246,126,230,92,200,50,106,118,3
    3
7290 DATA 11,118,205,201,116,195,45,64,33,50,118,24,245
7300 DATA 80,85,82,71,69,32,85,84,73,76,73,84,89,32,32,
    32,32
7310 DATA 32,86,69,82,83,73,79,78,32,49,46,48,13,13,0,3
    2
7320 DATA 32,32,32,32,32,32,32,32,32,32,32,0,32,32,32,8
    0
7330 DATA 85,82,71,69,32,40,89,47,78,41,32,32,0,42,32,6
    8
7340 DATA 69,76,69,84,73,78,71,32,58,32,0,13,68,73,82,6
    9
7350 DATA 67,84,79,82,89,32,82,69,65,68,32,69,82,82,79,
    82
7360 DATA 32,45,32,80,85,82,71,69,32,65,66,79,82,84,69,
    68
7370 DATA 13,0,13,68,82,73,86,69,32,78,79,84,32,82,69,6
    5
7380 DATA 68,89,32,45,32,80,85,82,71,69,32,65,66,79,82,
    84
7390 DATA 69,68,13,0,80,85,82,71,69,32,68,82,73,86,69,3
    2
7400 DATA 78,79,46,32,63,32,0
```

Program Listing 2 in Assembly Language

```
                 ;
                 ;      *******************************************************
                 ;      **             P U R G E   U T I L I T Y             **
                 ;      **                                                   **
                 ;      **          PURGE/DELETE DISKETTE FILES              **
                 ;      **              TRSDOS -OR- NEWDOS                   **
                 ;      **                                                   **
                 ;      ** AUTHOR     JOE LIGORI                             **
                 ;      **            2660 W. BALL RD.  # 60                 **
                 ;      **            ANAHEIM, CALIFORNIA   92804            **
                 ;      *******************************************************
                 ;
                 ;      ** --- VARIABLES --- **
                 ;
402D             DOS     EQU    402DH        ;DOS RETURN POINT
442C             KILL    EQU    442CH        ;TRSDOS KILL A FILE
4424             OPEN    EQU    4424H        ;TRSDOS OPEN A FILE
6300             IODCB   EQU    6300H        ;I/O CONTROL BLOCK
033A             CRT     EQU    033AH        ;TRS DISPLAY A CHARACTER
0049             KBD     EQU    0049H        ;TRS GET A CHARACTER KBD
01C9             CLS     EQU    01C9H        ;TRS CLS ROUTINE
6400             DBUF    EQU    6400H        ;DIRECTORY BUFFER
                 ;
7400                     ORG    7400H        ;29676 DECIMAL
7400             PURGE   EQU    $
7400 F3                  DI                  ;DISABLE INTERUPTS
7401 CDC901              CALL   CLS          ;CLEAR SCREEN
7404 210F75              LD     HL,SIGNON    ;POINT TO STRING
7407 CDC974              CALL   STRING       ;DISPLAY STING
740A 215476              LD     HL,GETDR     ;POINT TO 'DRIVE # ?'
740D CDC974              CALL   STRING       ;DISPLAY STRING
7410 CD4900      GDR     CALL   KBD          ;GET CHAR
7413 FE34                CP     '4'
7415 30F9                JR     NC,GDR       ;GO IF > 3
7417 FE30                CP     '0'
7419 30F5                JR     C,GDR        ;GO IF < 0
741B CD3A03              CALL   CRT          ;DISPLAY DRIVE #
741E D630                SUB    30H          ;MAKE HEX FROM ASCII
7420 3C                  INC    A            ;UP DRIVE #
7421 326776              LD     (DRIVE),A    ;STORE DR #
7424 CD6275              CALL   DELSEL       ;SELECT DRIVE W/DELAY
7427 CDD574              CALL   DIREAD       ;READ THE DIRECTORY
                 ;
                 ;      ** -- DISPLAY FILES AND GET RESPONSE -- **
                 ;
742A 0640        GETDIR  LD     B,40H        ;FILE COUNT
742C 210064              LD     HL,DBUF      ;POINT TO DIR BUFF
742F CD3E75              CALL   CRLF         ;CARRIAGE RET/LINE FEED
7432 C5          RD1     PUSH   BC           ;SAVE COUNTER
7433 E5                  PUSH   HL           ;SAVE POINTER
7434 CB76                BIT    6,(HL)       ;SYSTEM FILE ?
7436 2025                JR     NZ,RD10      ;GO IF SYSTEM FILE
7438 CB66                BIT    4,(HL)       ;VALID DIRECTORY ENTRY ?
743A 2821                JR     Z,RD10       ;GO IF NOT VALID
743C 110500              LD     DE,05H       ;   INCREMENT
743F 19                  ADD    HL,DE        ;   TO FILENAME
7440 CD1675              CALL   FNAME        ;DISPLAY FILENAME
7443 21EC75              LD     HL,PYN       ;PURGE (Y/N) ?
7446 CDC974              CALL   STRING       ;DISPLAY
7449 CDF474              CALL   GETYN        ;GET Y/N RESPONSE
744C FE59                CP     'Y'          ;DELETE ?
744E F5                  PUSH   AF           ;SAVE FLAG
744F CD3A03              CALL   CRT          ;DISPLAY Y-N
7452 CD3E75              CALL   CRLF         ;CARRIAGE RETURN
7455 F1                  POP    AF           ;GET FLAG
7456 E1                  POP    HL           ;RESTORE POINTER
7457 2003                JR     NZ,RD9       ;GO IF NO DELETE
7459 3EFF                LD     A,0FFH       ;DELETE MARKER = FF
745B 77                  LD     (HL),A       ;MARK IT TO DELETE
745C E5          RD9     PUSH   HL           ;BALANCE STACK
745D E1          RD10    POP    HL           ;RESTORE POINTER
745E 112000              LD     DE,20H       ;INC TO NEXT ENTRY
7461 19                  ADD    HL,DE        ;POINT TO NEXT
7462 C1                  POP    BC           ;COUNTER BACK
7463 10CD                DJNZ   RD1          ;READ ANOTHER
7465 CDC901              CALL   CLS          ;DONE HERE NOW DELETE
                 ;
                 ;      ** -- DELETE THE SPECIFIED FILES -- **
                 ;
7468 0640                LD     B,40H        ;# OF POSSIBLE FILES
746A 210064              LD     HL,DBUF      ;POINT TO BUFFER
746D C5          DD1     PUSH   BC           ;SAVE COUNTER
746E E5                  PUSH   HL           ;SAVE POINTER
```

Program continues

✓See List of Advertisers on page 306

```
746F 3EFF     00930        LD    A,0FFH        ;DELETE MARKER
7471 BE       00940        CP    (HL)          ;DELETE THIS ONE ?
7472 2047     00950        JR    NZ,DD10       ;GO IF NO DELETE
7474 CD4475   00960        CALL  FILL          ;FILL DCB W/ SPACES
7477 110500   00970        LD    DE,05H        ;     INCREMENT
747A 19       00980        ADD   HL,DE         ;      TO NAME
747B 110063   00990        LD    DE,IODCB      ;POINT TO DCB
747E 0608     01000        LD    B,08          ;8 CHR. NAME COUNTER
7480 7E       01010 DD3    LD    A,(HL)        ;GET CHARACTER
7481 FE20     01020        CP    ' '           ;SPACE ?
7483 2802     01030        JR    Z,DD4         ;GO IF A SPACE
7485 12       01040        LD    (DE),A        ;MOVE TO DCB
7486 13       01050        INC   DE            ;BUMP DCB POINTER
7487 23       01060 DD4    INC   HL            ;BUMP NAME POINTER
7488 10F6     01070        DJNZ  DD3           ;GO FOR NEXT IF > 0
748A 3E2F     01080        LD    A,'/'
748C 12       01090        LD    (DE),A        ;INSERT A / IN DCB
748D 13       01100        INC   DE            ;BUMP DCB POINTER
748E 0603     01110        LD    B,3           ;3 CHR EXTENSION COUNTER
7490 7E       01120 DD5    LD    A,(HL)        ;GET CHARACTER
7491 FE20     01130        CP    ' '           ;SPACE ?
7493 2805     01140        JR    Z,DD6         ;GO IF SPACE
7495 12       01150        LD    (DE),A        ;MOVE TO DCB
7496 23       01160        INC   HL            ;BUMP NAME POINTER
7497 13       01170        INC   DE            ;BUMP DCB POINTER
7498 10F6     01180        DJNZ  DD5           ;GO IF > 0
749A 3E0D     01190 DD6    LD    A,0DH         ;C/R
749C 12       01200        LD    (DE),A        ;END DCB WITH C/R
749D 21FD75   01210        LD    HL,PRG        ;POINT TO 'DELETING :'
74A0 CDC974   01220        CALL  STRING        ;DISPLAY 'DELETING : '
74A3 E1       01230        POP   HL            ;GET BEGINING POINTER
74A4 E5       01240        PUSH  HL            ;RE-SAVE POINTER
74A5 110500   01250        LD    DE,05         ;INCREMENT TO NAME
74A8 19       01260        ADD   HL,DE         ;POINT TO NAME
74A9 CD1675   01270        CALL  FNAME         ;DISPLAY FILENAME
74AC CD3E75   01280        CALL  CRLF          ;CAR RET / LINEFEED
74AF 21007E   01290        LD    HL,7E00H      ;POINT TO DUMMY BUFFER
74B2 110063   01300        LD    DE,IODCB      ;POINT TO DCB
74B5 CD2444   01310        CALL  OPEN          ;OPEN THE FILE
74B8 CD2C44   01320        CALL  KILL          ;KILL THE FILE
74BB E1       01330 DD10   POP   HL            ;GET POINTER
74BC 112000   01340        LD    DE,20H        ;     INCREMENT
74BF 19       01350        ADD   HL,DE         ;      TO NEXT ENTRY
74C0 C1       01360        POP   BC            ;RESTORE COUNTER
74C1 10AA     01370        DJNZ  DD1           ; GO IF > 0
74C3 CDC901   01380 BDOS   CALL  CLS           ;CLEAR SCREEN
74C6 C32D40   01390        JP    DOS           ;BACK TO  DOS READY'
              01400 ;
              01410 ;      ** -- SUBROUTINES -- **
              01420 ;
74C9          01430 STRING EQU   $             ;DISPLAY A STRING
74C9 7E       01440        LD    A,(HL)        ;GET CHARACTER
74CA FE00     01450        CP    0             ;END OF STRING ?
74CC C8       01460        RET   Z             ;GO IF END OF STRING
74CD E5       01470        PUSH  HL            ;SAVE STRING POINTER
74CE CD3A03   01480        CALL  CRT           ;CALL THE DISPLAY ROUTINE
74D1 E1       01490        POP   HL            ;RECOVER STRING POINTER
74D2 23       01500        INC   HL            ;BUMP POINTER
74D3 18F4     01510        JR    STRING        ;DO IT AGAIN
              01520 ;
74D5          01530 DIREAD EQU   $             ;READ THE DIRECTORY
74D5 3E11     01540        LD    A,11H         ;TRACK 17
74D7 326876   01550        LD    (TRACK),A     ;LOAD TRACK #
74DA 3E02     01560        LD    A,02H         ;START SECTOR 2
74DC 326976   01570        LD    (SECTOR),A    ;LOAD SECTOR #
74DF 010064   01580        LD    BC,DBUF       ;LOAD BC W/DIR BUFFER
74E2 CD6E75   01590 DIR1   CALL  SREAD         ;READ A SECTOR OF DIRECT.
74E5 DD216976 01600        LD    IX,SECTOR     ;POINT TO SECTOR STORAGE
74E9 DD3400   01610        INC   (IX)          ;UP SECTOR NO.
74EC 3E0A     01620        LD    A,0AH         ;STORE 10D
74EE DDBE00   01630        CP    (IX)          ;AT END OF TRACK ?
74F1 20EF     01640        JR    NZ,DIR1       ;GO IF MORE TO READ
74F3 C9       01650        RET                 ;END OF DIR READ
              01660 ;
74F4          01670 GETYN  EQU   $             ;GET A Y OR N RESPONSE
74F4 CD4900   01680        CALL  KBD           ;GET A CHARACTER
74F7 FE59     01690        CP    'Y'
74F9 C8       01700        RET   Z             ;RETURN IF A   Y
74FA FE4E     01710        CP    'N'
74FC C8       01720        RET   Z             ;RETURN IF AN   N
74FD FE58     01730        CP    'X'
74FF CAC374   01740        JP    Z,BDOS        ;TO DOS IF BREAK
7502 18F0     01750        JR    GETYN         ;GO IF INVALID CHAR
              01760 ;
              01770 ;    ROUTINES 'CRT1' AND 'CRT2' ARE TO INSERT FILENAMES
              01780 ;    INTO A SPACE FILLED STORAGE AREA FOR JUSTIFIED
              01790 ;    TEXT DISPLAYS.
              01800 ;
7504 7E       01810 CRT1   LD    A,(HL)        ;GET CHARACTER
7505 FE20     01820        CP    ' '           ;     SKIP IF
7507 2803     01830        JR    Z,CRTA        ;      A SPACE
7509 CD1075   01840        CALL  CRT2
750C 23       01850 CRTA   INC   HL            ;BUMP POINTER
750D 10F5     01860        DJNZ  CRT1          ;GO IF > 0
750F C9       01870        RET
              01880 ;
7510 DD7700   01890 CRT2   LD    (IX),A        ;INSERT CHARACTER
7513 DD23     01900        INC   IX            ;BUMP POINTER
7515 C9       01910        RET
              01920 ;
              01930 ;   THE 'FNAME' ROUTINE EXTRACTS A FILENAME AND
              01940 ;   DISPLAYS IT IN A JUSTIFIED MODE
              01950 ;
7516 060C     01960 FNAME  LD    B,0CH         ;12 CHARS MAX
7518 DD21DF75 01970        LD    IX,FNM        ;JUSTIFY AREA
751C CD4A75   01980        CALL  FILL2         ;FILL WITH SPACES
751F DD21DF75 01990        LD    IX,FNM
7523 0608     02000        LD    B,08
7525 CD0475   02010        CALL  CRT1
7528 3E20     02020        LD    A,' '
752A BE       02030        CP    (HL)
752B 280A     02040        JR    Z,FNAME2
752D 3E2F     02050        LD    A,'/'
752F CD1075   02060        CALL  CRT2
7532 0603     02070        LD    B,3
7534 CD0475   02080        CALL  CRT1
7537 21DF75   02090 FNAME2 LD    HL,FNM
753A CDC974   02100        CALL  STRING
753D C9       02110        RET
              02120 ;
753E 3E0D     02130 CRLF   LD    A,0DH         ;CARRIAGE RET / LINE FEED
7540 CD3A03   02140        CALL  CRT           ;DISPLAY IT
7543 C9       02150        RET
              02160 ;
7544 0620     02170 FILL   LD    B,20H         ;FILL AREA WITH 20 SPACES
```

Program continues

```
7546 DD210063 02180          LD      IX,IODCB
754A DD360020 02190 FILL2    LD      (IX),' '
754E DD23     02200          INC     IX
7550 10F8     02210          DJNZ    FILL2
7552 C9       02220          RET
              02230 ;
              02240 ;    SELECT A DISK DRIVE
              02250 ;
7553 C5       02260 SEL      PUSH    BC                 ;SAVE BC
7554 3A6776   02270          LD      A,(DRIVE)          ;GET DRIVE #
7557 47       02280          LD      B,A                ;INTO B FOR COUNTER
7558 3E80     02290          LD      A,80H              ;SET BITS FOR SELECTION
755A 07       02300 SEL1     RLCA                       ;ROTATE
755B 10FD     02310          DJNZ    SEL1               ; UNTIL IN POSITION
755D 32E137   02320          LD      (37E1H),A          ;SELECT THE DRIVE
7560 C1       02330          POP     BC                 ;RESTORE BC
7561 C9       02340          RET
              02350 ;
              02360 ;    SELECT A DISK DRIVE WITH A DELAY FOR 1ST ACCESS
              02370 ;
7562 CD5375   02380 DELSEL   CALL    SEL                ;SELECT THE DRIVE
7565 210000   02390          LD      HL,00              ;ZERO HL
7568 2B       02400 DLY      DEC     HL
7569 7C       02410          LD      A,H                ;DELAY ROUTINE
756A B5       02420          OR      L
756B 20FB     02430          JR      NZ,DLY
756D C9       02440          RET
              02450 ;
              02460 ;** SECTOR READ ROUTINE **
              02470 ;        ENTER WITH BC=BUFFER ADDRESS
              02480 ;        DRIVE, TRACK, SECTOR ARESTORED IN MEMORY
              02490 ;
756E CD5375   02500 SREAD    CALL    SEL                ;SELECT DRIVE
7571 3A6876   02510          LD      A,(TRACK)          ;GET TRACK #
7574 32EF37   02520          LD      (37EFH),A          ;SELECT TRACK
7577 3A6976   02530          LD      A,(SECTOR)         ;GET SECTOR #
757A 32EE37   02540          LD      (37EEH),A          ;SELECT SECTOR
757D 21EC37   02550          LD      HL,37ECH           ;POINT TO STAT/COMM REG
7580 3E1E     02560          LD      A,1EH              ;SEEK COMMAND
7582 77       02570          LD      (HL),A             ;-SEEK-
7583 C5       02580          PUSH    BC
7584 C1       02590          POP     BC                 ;WASTE
7585 C5       02600          PUSH    BC                 ;TIME
7586 C1       02610          POP     BC
7587 7E       02620 CREADY   LD      A,(HL)             ;GET STATUS
7588 07       02630          RLCA
7589 3003     02640          JR      NC,CBUSY           ;GO IF READY
758B C3BA75   02650          JP      NREADY             ;DRIVE NOT READY
758E 0F       02660 CBUSY    RRCA                       ;RE-ADJUST BYTE
758F 0F       02670          RRCA                       ;BUSY BIT TO FLAG
7590 38F5     02680          JR      C,CREADY           ;GO IF BUSY
7592 368B     02690 READ     LD      (HL),88H           ;GIVE READ COMMAND
7594 11EF37   02700          LD      DE,37EFH           ;POINT TO DATA REGISTER
7597 C5       02710          PUSH    BC
7598 C1       02720          POP     BC                 ;WASTE
7599 C5       02730          PUSH    BC                 ;TIME
759A C1       02740          POP     BC
759B 1803     02750          JR      READ2
759D 0F       02760 READ1    RRCA                       ;BUSY BIT TO FLAG
759E 300A     02770          JR      NC,STATCK          ;BUSY/DONE GO CK STATUS
75A0 7E       02780 READ2    LD      A,(HL)             ;GET STATUS BYTE
75A1 CB4F     02790          BIT     1,A                ;DRQ BIT
75A3 28F8     02800          JR      Z,READ1            ;GO CHECK BUSY
75A5 1A       02810 READIT   LD      A,(DE)             ;GET DATA BYTE
75A6 02       02820          LD      (BC),A             ;STORE TO BUFFER
75A7 03       02830          INC     BC                 ;BUMP BUFFER POINTER
75A8 18F6     02840          JR      READ2              ;GO CHK DRQ
75AA 7E       02850 STATCK   LD      A,(HL)             ;GET STATUS BYTE
75AB E65C     02860          AND     5CH                ;MASK IT
75AD C8       02870          RET     Z                  ;GO IF NO ERROR
75AE 326A76   02880          LD      (ERCODE),A         ;STORE THE ERROR BYTE
75B1 210B76   02890          LD      HL,RERR            ;'DIRECTORY READ ERROR'
75B4 CDC974   02900 DEROR    CALL    STRING             ;DISPLAY MSG
75B7 C32D40   02910          JP      DOS                ;ABORT
75BA 213276   02920 NREADY   LD      HL,NRERR           ;'DRIVE NOT READY'
75BD 18F5     02930          JR      DEROR              ;DISPLAY & ABORT
              02940 ;
              02950 ;        ** -- MESSAGES / DISPLAYS -- **
              02960 ;
75BF          02970 SIGNON   EQU     $
75BF 50       02980          DEFM    'PURGE UTILITY     VERSION 1.0'
     55 52 47 45 20 55 54 49
     4C 49 54 59 20 20 20 20
     20 56 45 52 53 49 4F 4E
     20 31 2E 30
75DC 0D0D     02990          DEFW    0D0DH
75DE 00       03000          DEFB    00H
              03010 ;
75DF 20       03020 FNM      DEFM    '           '
     20 20 20 20 20 20 20 20
     20 20 20
75EB 00       03030          DEFB    00
              03040 ;
75EC 20       03050 PYN      DEFM    '   PURGE (Y/N)  '
     20 20 50 55 52 47 45 20
     28 59 2F 4E 29 20 20
75FC 00       03060          DEFB    00
75FD 2A       03070 PRG      DEFM    '* DELETING : '
     20 44 45 4C 45 54 49 4E
     47 20 3A 20
760A 00       03080          DEFB    00
              03090 ;
760B 0D       03100 RERR     DEFB    0DH
760C 44       03110          DEFM    'DIRECTORY READ ERROR - PURGE ABORTED'
     49 52 45 43 54 4F 52 59
     20 52 45 41 44 20 45 52
     52 4F 52 20 2D 20 50 55
     52 47 45 20 41 42 4F 52
     54 45 44
7630 0D00     03120          DEFW    000DH
              03130 ;
7632 0D       03140 NRERR    DEFB    0DH
7633 44       03150          DEFM    'DRIVE NOT READY - PURGE ABORTED'
     52 49 56 45 20 4E 4F 54
     20 52 45 41 44 59 20 2D
     20 50 55 52 47 45 20 41
     42 4F 52 54 45 44
7652 0D00     03160          DEFW    000DH
              03170 ;
7654 50       03180 GETDR    DEFM    'PURGE DRIVE NO. ? '
     55 52 47 45 20 44 52 49
     56 45 20 4E 4F 2E 20 3F
     20
7666 00       03190          DEFB    00H
              03200 ;
0001          03210 DRIVE    DEFS    01                 ;1 BYTE DRIVE STORAGE
0001          03220 TRACK    DEFS    01                 ;1 BYTE TRACK STORAGE
0001          03230 SECTOR   DEFS    01                 ;1 BYTE SECTOR STORAGE
```


✓See List of Advertisers on page 306

Save time by purging yourself of redundant activities.

Copykill

Steve Kelley
9506 Peach St.
Oakland, CA 94603

Copykill will selectively copy or kill dozens of programs with one command, zap new names on your diskettes and sort and LPRINT alphabetical listings of all your diskette directories. All this is done without entering program names or data manually.

The tricks and patches I used writing Copykill are detailed and explained for use in other programs.

This is the type of program that just about everybody thinks of writing, but no one ever finds the time to do.

These programs won't balance your checkbook or destroy any Klingons, but you'll find a few in almost every program library. Utility programs—programs that really don't do anything practical or useful by themselves— provide an easier way to accomplish what you wanted to do in the first place. If you're like me, you know what it's like to sit down at the computer ready to write the world's greatest, universal, do-all program, only to end up spending hours trying to work around the shortcomings of the computer.

Utility programs can help by turning the computer into a better, more capable tool. If you spend all your time writing utilities, however, you'll never have time for the program you first set out to write. What's worse is that most of your time will be spent reinventing the wheel—writing utilities that already exist and trying out ideas that have already been tried.

Throughout my experience with microcomputers I have developed ideas and tricks to make things run easier, or faster, or maybe just better. Some of these would prove helpful to anyone writing programs for their TRS-80. Copykill presents some of these ideas, demonstrating its implementation, as well as its use. The program containing these routines is a very handy utility as well. The more wheels you don't have to reinvent, the better, right?

```
********************************
        COPYKILL MENU
********************************

1 = COPY  - COPY ANY OR ALL DISK FILES
2 = PURGE - KILL ANY OR ALL DISK FILES
3 = LPRINT DISK DIRECTORIES
4 = SORT & LIST DISK DIRECTORIES
5 = EDIT DISK ID'S
6 = DOS COMMAND
7 = LPRINT FORM FEED

    WHICH ONE ? _
```

Table 1

FILE DIRECTORY — NDOS 21A - 01/03/80 1 GRANS FREE

PENCIL/CMD	TESTDAT2	TEST2/CMD	HEX2ASC
LP3/CMD	COPYKILL	DISKDUMP/NEW	DISASSEM/CMD
SUPERZAP	BASIC/CMD	FORMAT/CMD	LEVEL1/CMD
BACKUP/CMD	COPY/CMD	SUPERZAP/COM	LMOFFSET/CMD
LV1CONV/CMD	DIRCHECK/CMD	REM/OBJ	DAYOWEEK
SUPEREDT			

FILE DIRECTORY — DATA 3B - 04/10/80 14 GRANS FREE

SARGON/COM	CHKBOOK	TICTACTO	SARGON/CMD
ENDZONE	STARMIKE	BIORYTHM	DISKEDIT
LUNARLND	COMPOSER	KENO2	LANDER
LUNARLND/CMD	SLOTS	BRIDGE	AIRAID/CMD
KENO1		REM/OBJ	DAYOWEEK

FILE DIRECTORY — DATA 5A - 03/28/79 19 GRANS FREE

SUPERTREK	SAILOR1	TANKWAR	SAILOR2
BACGAMMN	DIETST2/TXT	BANNER	SMARTTAC
EXPTEST	PILLBOX	UART/CMD	DIETPLAN
INSTR	BLACKJCK	SWITCH	HARRY2
PICTURE	SPACEGUN	UART/OBJ	DIETAIDS
UART	DIETST1/TXT		

Table 2. Sample print-out of LPRINT DIRECTORY command.

Tricks and Ideas

How many times have you attempted to copy all your programs onto disks in an organized manner, and create a catalog so they would be easy to find later? How many times have you succeeded? Answer A minus answer B will yield the approximate Excedrin headache number.

Each time I tried, I would get about halfway through, get bored, and stop. This only compounded my problems by leaving the original programs in random order on the same number of disks, adding a few organized but unlabeled disks to the pile, and all with no list. I finally got fed up not being able to find anything, and decided to write a utility program to copy programs from disk to disk and kill duplicate programs without having to enter their names or even the words copy or kill. More functions were added later.

Operation

The copy function (see menu in Table 1) will read the directory of the source drive and prompt a Y or N (yes or no) input for each entry. Completing this, Copykill copies all selected programs to the destination drive. If the source and destination drives are the same, temporary files are opened on drive zero to contain all files to be copied. Then only one disk swap is necessary regardless of the number of programs being copied.

Copykill's purge function is almost as useful as Copy in saving time and work. After you copy 23 games from your financial disk to your amusement disk with a single command, you still must go back and kill them to free the space. Purge reads the source drive's directory and prompts Y or N for each entry just as copy does. Then as an extra precaution, it lists all the files to be killed on the screen and asks if you are sure. If you are, each selected file is displayed and killed. The Copykill program uses this routine to kill all temporary files after a one drive copy.

Finally, all my disks are organized. Now how about that list I needed?

The LPrint Directory routine was originally written for that. It LPRINTs each disk's directory in neat columns along with the disk's ID and free space (see Table 2.) It does so as fast as you can feed the disks into the drive to be read. This enables you to see what's on a disk and is certainly faster than doing directory reads of 25 disks to find the program you want. If you're looking for a particular program, you still have to search a bunch of random lists; that's why I added the Sort and List function. It lets you feed in as many disks as you want; sorts the list into alphabetical order; and yields a printout, shown in Table 3. Notice that the entries are in order vertically, so column one contains entries from A to G, column two contains H to N, etc. I find this format easier to

PROGRAM	DISK ID	PROGRAM	DISK ID	PROGRAM	DISK ID
AIRAID/CMD	DATA 3B	DISKEDIT	DATA 3B	SAILOR1	DATA 5A
BACGAMN	DATA 5A	ENDZONE	DATA 3B	SAILOR2	DATA 5A
BACKUP/CMD	NDOS 21A	EXPTEST	DATA 5A	SARGON/CMD	DATA 3B
BANNER	DATA 5A	FORMAT/CMD	NDOS 21A	SARGON/COM	DATA 3B
BASIC/CMD	NDOS 21A	HARRY2	DATA 5A	SLOTS	DATA 3B
BIORYTHM	DATA 3B	HEX2ASC	NDOS 21A	SMARTTAC	DATA 5A
BLACKJCK	DATA 5A	INSTR	DATA 5A	SPACEGUN	DATA 5A
BRIDGE	DATA 3B	KEN01	DATA 3B	STARMIKE	DATA 3B
CHKBOOK	DATA 3B	KEN02	DATA 3B	SUPEREDT	NDOS 21A
COMPOSER	DATA 3B	LANDER	DATA 3B	SUPERZAP	NDOS 21A
COPY/CMD	NDOS 21A	LEVEL1/CMD	NDOS 21A	SUPERZAP/COM	NDOS 21A
COPYKILL	NDOS 21A	LMOFFSET/CMD	NDOS 21A	SUPRTREK	DATA 5A
DAYOWEEK	NDOS 21A	LP3/CMD	NDOS 21A	SWITCH	DATA 5A
DIETAIDS	DATA 5A	LUNARLND	DATA 3B	TANKWAR	DATA 5A
DIETPLAN	DATA 5A	LUNARLAND/CMD	DATA 3B	TEST2/CMD	NDOS 21A
DIETST1/TXT	DATA 5A	LV1CONV/CMD	NDOS 21A	TESTDAT2	NDOS 21A
DIETST2/TXT	DATA 5A	PENCIL/CMD	NDOS 21A	TICTACTO	DATA 3B
DIRCHECK/CMD	NDOS 21A	PICTURE	DATA 5A	UART	DATA 5A
DISASSEM/CMD	NDOS 21A	PILLBOX	DATA 5A	UART/CMD	DATA 5A
DISKDUMP/NEW	NDOS 21A	REM/OBJ	NDOS 21A	UART/OBJ	DATA 5A

Table 3. Sample print-out of SORT and LIST command.

Program Listing

```
10 '                  ***       COPYKILL          ***
20 CLEAR 16800
30 DEFINT P-Z:DEFSTR A-H:DEFDBL M
40 DEF FNA(X)=MID$(STR$(X),2,10)
50 '
60 '    ***       CHECK FOR DATA STATEMENT ERRORS        ***
70 '    ***      (TO BE REMOVED AFTER INITIAL RUN)       ***
80 M=0:FOR Z=1 TO 175:READ Z1:M=M+Z1:NEXT Z:RESTORE
90 IF M<>364959 THEN PRINT"DATA STATEMENT ERROR !!":STO
   P
100 '
110 '     ***       PATCH FOR FAST CMD "DOS COMMAND"      **

120 FOR Z=0 TO 6:READ W1(Z):W2(Z)=PEEK(W1(Z)):NEXT Z
130 DATA 21690,21691,21692,21729,21730,21731,21732
140 '
150 '     ***       PATCH FOR READING DIR & FREE        ***
```

Program continues

```
160 DIM F(35),R(35),F1(800),F2(35),H(40)
170 FOR Z=0 TO 40:READ Z1,Z2,Z3:K4=VARPTR(H(Z)):POKE K4
     ,Z1
180 K4=K4+1:IF K4>32767 THEN K4=K4-65536
190 POKE K4,Z2:K4=K4+1:IF K4>32767 THEN K4=K4-65536
200 POKE K4,Z3:NEXT Z
210 DATA 13,128,60
220 DATA 13,148,60,13,168,60,13,192,60,13,212,60,13,232
     ,60
230 DATA 13,0,61,13,20,61,13,40,61,13,64,61,13,84,61
240 DATA 13,104,61,13,128,61,13,148,61,13,168,61,13,192
     ,61
250 DATA 13,212,61,13,232,61,13,0,62,13,20,62,13,40,62
260 DATA 13,64,62,13,84,62,13,104,62
270 DATA 13,128,62,13,148,62,13,168,62,13,192,62,13,212
     ,62
280 DATA 13,232,62,13,0,63,13,20,63,13,40,63,13,64,63
290 DATA 13,84,63,13,104,63
300 DATA 21,31,60,2,49,60,2,113,60,2,177,60,2,241,60
310 '
320 '            ***     PATCH FOR FASTER SORT      ***
330 DIM S(200)    '  ROOM FOR MACHINE LANGUAGE PART
340 FOR Z=0 TO 9:READ S(Z):NEXT Z
350 DATA 8448,0,4352,0,774,19994,31095,4882,4131,-13833
360 '
370 '            ***     PATCH FOR DISK I/O     ***
380 DIM P(250)    ' ROOM FOR MACHINE LANGUAGE ROUTINE
390 R1=35   'BUFFER STARTS HERE
400 FOR Z=0 TO R1-1:READ P(Z):NEXT Z
410 DATA -6691,8669,0,8448,15360,30173,-8957,1140,-6691
420 DATA 465,0,17101,8260,-13053,17462,9839,-6960,-1539
     1
430 DATA 2714,24704,17920,252,2049,0,0,350,7936,32,7952
440 DATA 64,7968,96,7984,128,8000
450 '
460 CLS:INPUT"HOW MANY DRIVES TOTAL  ( 1-4 )";QD:QD=QD-
     1
470 IF QD<0 OR QD>3 THEN 460
480 FOR X=0 TO 35:F2(X)="TEMP"+FNA(X)+":0":NEXT X
490 '            ***     MENU      ***
500 FOR Z=0 TO 6:POKE W1(Z),W2(Z):NEXT Z ' * RESTORE DE
     LAY *
510 CLS:PRINT,STRING$(32,"*")
520 PRINT,"           COPYKILL MENU"
530 PRINT,STRING$(32,"*"):PRINT
540 PRINT,"1 = COPY  - COPY ANY OR ALL DISK FILES"
550 PRINT,"2 = PURGE - KILL ANY OR ALL DISK FILES"
560 PRINT,"3 = LPRINT DISK DIRECTORIES"
570 PRINT,"4 = SORT & LIST DISK DIRECTORIES"
580 PRINT,"5 = EDIT DISK ID'S"
590 PRINT,"6 = DOS COMMAND"
600 PRINT,"7 = LPRINT FORM FEED"
610 'ADD YOUR OWN COMMANDS HERE
620 PRINT:PRINT,"    WHICH ONE ";:INPUT R:IF R<1 THEN 5
     00
630 FOR Z=0 TO 6:POKE W1(Z),0:NEXT Z
640 ON R GOTO 670,1130,1360,1520,1720,2030,2090
650 GOTO 500
660 '
670 '            ***     COPY    ***
680 CLS:IF QD=0 THEN PS=0:PD=0:GOTO 720
690 GOSUB 2270
700 INPUT"DESTINATION DRIVE";PD:IF PD<0 OR PD>QD THEN 7
     00
710 PD$=":"+FNA(PD)
720 IF PS<>PD OR PS>0 THEN 740
730 INPUT"NO 1 DRIVE COPIES ON DRIVE ZERO !!";C:GOTO 50
     0
740 IF PS<>PD THEN 760
750 PRINT"HIT  <ENTER>  WHEN SOURCE DISK IS IN DRIVE";P
     S;:INPUT C
760 GOSUB 2130:IF U=-1 THEN 500
770 CLS:Q9=0:FOR X=0 TO U
780   F1(X)=F(X)+PD$:F(X)=F(X)+PS$
790   C="":PRINT"COPY "+F(X)+"   (Y/N)";:INPUT C
800   R(X)=0:IF C="Y" THEN R(X)=1:Q9=1
810 NEXT X
820 IF Q9=0 THEN 500
830 C="":CLS:INPUT"CHANGE ANY DESTINATION FILESPECS  (Y
     /N)";C
840 IF C<>"Y" THEN 920
850 CLS:PRINT"HIT  <ENTER>  TO LEAVE UNCHANGED.":PRINT
860 FOR X=0 TO U
870   IF R(X)<>1 THEN 910
880   PRINT F1(X)+" ? ";:LINE INPUT H
890   IF H<>"" THEN F1(X)=H
900   IF INSTR(F1(X),":")=0 THEN F1(X)=F1(X)+PD$
910 NEXT X
920 CLS:IF PS=PD THEN PRINT"MAKE SURE SOURCE IS STILL I
     N DRIVE";PS
930 C="":INPUT"READY TO COPY   (Y/N)";C
940 IF C<>"Y" THEN 500
950 IF PS<>PD THEN 1030
960 FOR X=0 TO U
970   IF R(X)<>1 THEN 1010
980   A$="COPY "+F(X)+" TO "+F2(X):PRINT A$;
990   CMD A$
1000   PRINT CHR$(27);:F(X)=F2(X)
```

Program continues

seerch quickly, and therefore worth the extra programming effort.

You may have noticed the unique disk IDs in the printout; where an ID would usually read simply NEWDOS or DATA, instead they read NDOS 21A or DATA 3B, for example. Having every disk with the seme ID seemed like a dumb idea, so I zapped my own IDs in their place. At first I used Superzap by Aparat to do this, but later I added the Edit Disk ID function to Copykill. I wouldn't then have to think in hex while working in ASCII. The letters DOS within the ID, of course, indicate that the disk will work in drive zero as opposed to data disks, which will not. The A and B in the ID number refer to sides A and B of the same disk, since I punch the extra sector hole and use both sides of each diskette. Now whenever a DIR is done, a unique disk name appears on the screen for visual indication and for Copykill to use in printouts.

I later added the remaining items to the menu for more functions. I indicated with REM statements where to add to the program and make the new functions appear on the menu. Some areas for improvement in the Copykill program might be to store the cumulative index on disk, and, maybe, add a comment field with some editing features to the alphabetized printout routine for program descriptions.

You may wish to change the LPRINT" " statements to LPRINT CHR$(138), or simply LPRINT depending on the type of printer you have. These will do a linefeed alone, instead of printing a space, carriage return, and linefeed, and speed up the printer operation.

If you have only one drive, or you're not using NEWDOS by Aparat, the copy function will not work. Without NEWDOS you will have to change the DIR statements to your DOS format.

Bear in mind that if your DOS is different, the POKEs in lines 60,140 and 260 will not work. You will have to change the DIM and Clear statements if you do not have 48K of memory, because they were set as large as possible for the Sort & List function of Copykill. Each program name stored requires one array element and 21 bytes of string space.

Obstecles

Here are some of the problems I had writing Copykill.

The following tricks led to their solution. First I wanted to copy programs from disk to disk without having to enter a COPY command for every program. That was easy enough to do, with Copykill using COPY etc. It took longer than I felt was necessary, though.

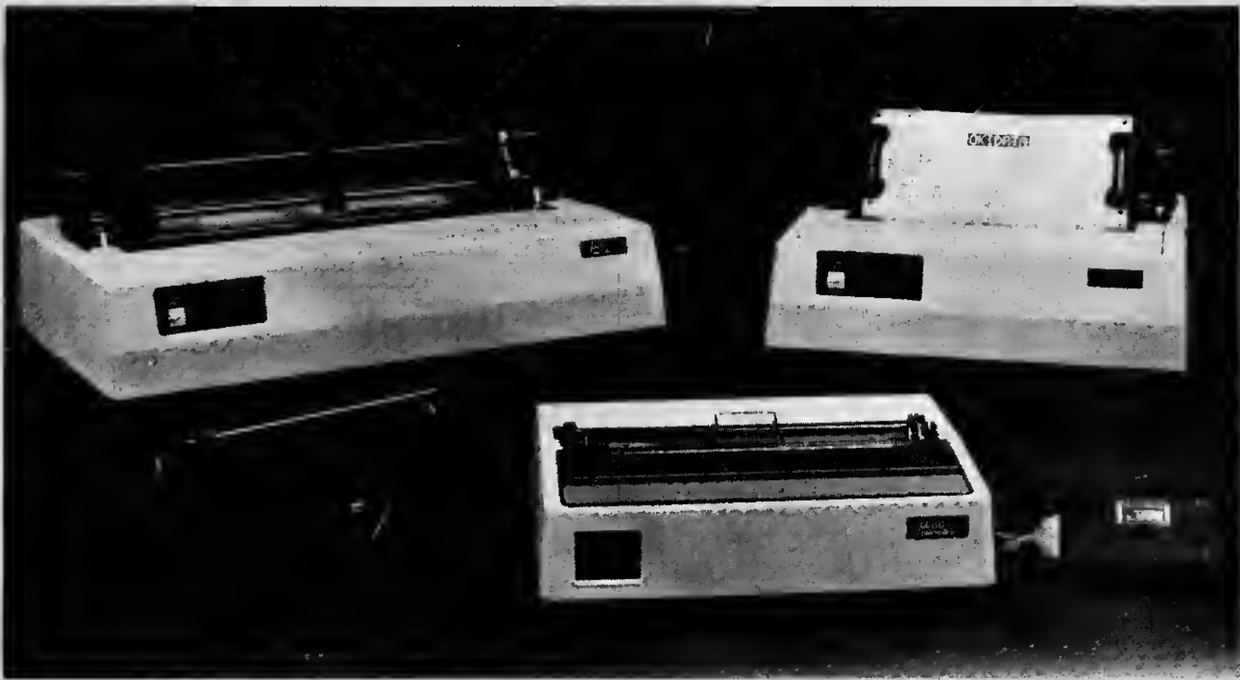
I needed a patch of some kind to get rid of the familiar delay when using WHATEVER, entering and leaving DOS from BASIC. After a lot of disassembly and searching, I found the answer. The POKEs in line 630 remove that delay, allowing instant access to DOS commands (Program Listing 1). This patch remains in effect until the system is rebooted, or, as I have chosen to do, the original code is POKEd back into place in line 500. This happens immediately on return from the DOS routine. This original code was PEEKed and stored for furture reference in line 120.

For the technically minded, these POKEs nullify two calls to a routine which generates a checksum of the BASIC interpreter before jumping to DOS. Then it regenerates the checksum upon return to BASIC to make sure it remains intact.

If it's different, the system reboots. Besides the annoying delay, this also prevents you from making intentional changes to BASIC should you ever desire to do so.

If you would like to remove these delays permanently, you can use SUPERZAP to modify BASIC/CMD on the disk. The areas to be changed are in BASIC/CMD, relative sector 3. That's the third sector from the beginning of BASIC/CMD (the first sector being zero). The first change is at MOD BA. What you should find is 41CD A453 which should be changed to 4100 0000. The second change is in the same sector, MOD E2. CDAE

OKIDATA Microlines...



at "OKidoki" Discounts!

Every little bit helps and we can help you save 3% for cash (in addition to our already low prices) on the Microline 80, 82 and 83 as well as several options.

What can we say that you don't already know? The Okidata Microline Series is a pacesetter for the industry and the Microline 80 is an absolute dream to use. (Microcomputing Magazine, Oct 80). Impressively small, it has a low mass print head that guarantees 200 million extremely clear characters in upper and lower case at 6 or 8 lines per inch! Standard 2" typewriter ribbons can be used and with an optional roll paper holder, inexpensive roll paper can be utilized. Microlines connect directly to TRS-80™, APPLE™ and other computers.

LTD ATA INC. 11

Compute **These** 3% Cash Savings!

	OUR REG. PRICE	CASH DISC. PRICE
MICROLINE 80	$489.00	**$474.33**
Tractors		
with Printer Purchase	62.00	**60.14**
without Printer Purchase	73.00	**70.81**
Roll Paper Holder		
with Printer Purchase	36.00	**34.92**
without Printer Purchase	42.00	**40.74**
RS-232	170.00	**164.90**
MICROLINE 82	649.00	**629.53**
Tractors	(same as for Microline 80)	
Roll Paper Holders	(same as for Microline 80)	
High Speed Buffered RS-232		
256 char.	180.00	**174.60**
2-K char.	264.00	**256.08**
MICROLINE 83	964.00	**935.08**
High Speed Buffered RS-232	(same as for Microline 82)	

SHIPPING... Via Standard UPS INCLUDED. Cash Discount honored with check, money order or COD's. We accept Visa or MasterCard, but cash discount does not apply. With credit card orders, include card number and expiration date.

C.O.D. shipments, $2.00 additional. Direct mail order to LT DATA INC. at either address below.

24 Hour phone Order Line 1-303-575-8518

P.O. Box 1567, Wheat Ridge, Colorado 80033
Colorado Residents add 3% Sales Tax
P.O. Box 1017, Medina, Ohio 44258
Ohio Residents add 5½% Sales Tax

NOTE: Verbatim Disks and Some TDK Cassettes still available at our low advertised prices (see April and May ads).

```
1010 NEXT X
1020 IF PS=PD INPUT"HIT  <ENTER>  WHEN DESTINATION DISK
      IN PLACE.";C
1030 FOR X=0 TO U
1040   IF R(X)<>1 THEN 1080
1050   A$="COPY "+F(X)+" TO "+F1(X):PRINT A$;
1060   CMD A$
1070   PRINT CHR$(27);
1080 NEXT X
1090 IF PS=PD THEN 1270
1100 INPUT"HIT  <ENTER>  TO CONTINUE";C
1110 GOTO 500
1120 '
1130 '             ***      PURGE     ***
1140 CLS:GOSUB 2270:GOSUB 2130:IF U=-1 THEN 500
1150 CLS:Q9=0:FOR X=0 TO U
1160   F(X)=F(X)+PS$
1170   C="":PRINT"KILL "+F(X)+"     (Y/N)";:INPUT C
1180   R(X)=0:IF C="Y" THEN R(X)=1:Q9=1
1190 NEXT X
1200 IF Q9=0 THEN 500
1210 CLS:Z=0:FOR X=0 TO U
1220   IF R(X)<>1 THEN 1250
1230   PRINT TAB(Z*32);"KILL "+F(X);
1240   Z=ABS(Z-1):IF Z=0 THEN PRINT
1250 NEXT X
1260 PRINT:C="":INPUT"ARE YOU SURE  (Y/N)";C:IF C<>"Y"
     THEN 500
1270 FOR X=0 TO U
1280   IF R(X)<>1 THEN 1320
1290   A$="KILL "+F(X):PRINT A$;
1300   KILL F(X)
1310   PRINT" - KILLED."
1320 NEXT X
1330 INPUT"HIT  <ENTER>  TO CONTINUE";B
1340 GOTO 500
1350 '
1360 '             ***      LPRINT DIRECTORIES        ***
1370 CLS:GOSUB 2270:GOTO 1400
1380 CLS:C="":INPUT"CONTINUE  (Y/N)";C
1390 IF C="N" THEN 500
1400 GOSUB 2130:LPRINT F
1410 IF U=-1 THEN 1490
1420 LPRINT" "
1430 FOR Z=0 TO U STEP 4
1440   FOR Z1=0 TO 3
1450     LPRINT TAB(Z1*20);F(Z+Z1);
1460   NEXT Z1
1470   LPRINT" "
1480 NEXT Z
1490 LPRINT" ":LPRINT" ":LPRINT" "
1500 GOTO 1380
1510 '
1520 '             ***      SORT & LIST      ***
1530 CLS:GOSUB 2270:Q1=0
1540 C="":CLS:INPUT"CONTINUE  (Y/N)";C:IF C="N" THEN 16
     10
1550 GOSUB 2130:IF U=-1 THEN 1540
1560 FOR X=0 TO U
1570   Q1=Q1+1
1580   F1(Q1)=F(X)+STRING$(13-LEN(F(X))," ")+LEFT$(H(36
     ),8)
1590 NEXT X
1600 GOTO 1540
1610 DEFUSR0=VARPTR(S(0))
1620 FOR Z1=1 TO Q1-1:FOR Z2=Q1 TO Z1+1 STEP -1
1630   IF F1(Z1)>F1(Z2) THEN S(1)=VARPTR(F1(Z1)):S(3)=V
     ARPTR(F1(Z2)):Z=USR0(Z)
1640 NEXT Z2:PRINT Q1-Z1;:NEXT Z1:Q2=Q1/3+.8
1650 FOR X=0 TO Q2:FOR Z=0 TO 2
1660  IF X=0 LPRINT TAB(Z*28);"PROGRAM       DISK ID";:G
     OTO 1690
1670  IF X=1 AND Z=0 LPRINT" ":LPRINT" "
1680  LPRINT TAB(Z*28);F1(X+Q2*Z);
1690 NEXT Z:LPRINT" ":NEXT X
1700 GOTO 500
1710 '
1720 '             ***      EDIT DISK ID     ***
1730 CLS:Q=0:GOSUB 2270
1740 TRACK=17:SECT=0:RW=0:GOSUB 1920
1750 A1="":FOR Z=R1+104 TO R1+107
1760   K=P(Z):IF K<0 THEN K=K+65536
1770   Z1=K/256:Z2=K-Z1*256:A1=A1+CHR$(Z2)+CHR$(Z1)
1780 NEXT Z
1790 PRINT Q;"DISK ID  ( IF ITS  ";
1800 PRINT CHR$(34)+A1+CHR$(34)+"  HIT  <ENTER> )  ?";
1810 LINE INPUT C
1820 IF C<>"" THEN A1=C ELSE GOTO 1910
1830 IF LEN(A1)<>8 THEN PRINT"MUST BE 8 CHARACTERS":GOT
     O 1750
1840 '          ***     WRITE DISK ID      ***
1850 FOR Z=0 TO 3
1860   K=ASC(MID$(A1,1+Z*2,1))+ASC(MID$(A1,2+Z*2,1))*25
     6
1870   IF K>32767 THEN K=K-65536
1880   P(Z+R1+104)=K
1890 NEXT Z
1900 TRACK=17:SECT=0:RW=1:GOSUB 1920
1910 C="":INPUT"CONTINUE   (Y/N) ";C:IF C="N" THEN 500
```

Program continues

53C2 should be changed to 0000 0000. Please note that these changes are for NEWDOS and wil *not* work with TRSDOS!

I also wanted to be able to read the disk directory into strings that a BASIC program could use for sorting or prompting. At first I displayed the directory on the video screen with DIR, then PEEKed at the screen where I knew the entries would be until I found an entry of all spaces.

The same applied for FREE. This produced the desired results, but again it seemed very slow. After some thought, I decided that a faster way would be to point the appropriate number of string pointers (found using VARPTR) at the screen. From then on, anything appearing on the screen in those locations would also be in the corresponding string. Fortunately, it's not as hard to do as it is to explain.

Line 160 sets up an array of strings called H, and lines 170—200 find the string pointers and POKE them to point to their respective screen locations. From there it's easy. Anything on the screen where the pointers point is also in that string. When the directory is displayed, the first entry on the screen is also in H(0), the second in H(1), etc. When FREE is entered, H(36) points to the free space on drive 0, H(37) to drive one, etc. As soon as the information appears on the screen these strings are transferred to another string array. The reason for this is that when the information disappears from the screen, it disappears from the H strings.

Finally, I wanted a faster sort than was available in BASIC. I didn't want, however, to have to protect memory and load a machine language file, or any of the other nuisances associated with machine language links to BASIC.

I tried all kinds of ways to avoid this "one program for the price of two" problem.

I tried POKEing the code into strings, which worked for small routines although there were problems passing arguments. I then realized if I were to use an integer array, I could load the routine directly from data statements and pass arguments to and from it just by knowing which array element that part of the code was in. The array could be DIMed to any size, too. The bubble sort itself is written in BASIC for simplicity, but the actual string manipulation is done in machine language. The code for the language part is loaded into the S integer array with the READ loop in line 340. Each string swap is called with USR0, with the location (VARPTR) of the appropriate strings in S(1) and S(3).

The speed advantages of the machine language routine are twofold. First, only the string pointers are swapped, so the strings themselves never have to be moved; second, since the strings are never moved, no temporary buffer strings are created by BASIC. BASIC's garbage collection routine, therefore, is never invoked. When sorting strings in a relatively full string space, this overhead time can account for more than 95 percent of sorting time.

I also wanted to change disk IDs. This involves writing a machine language routine to read and write disk sectors. It is not really a trick, but it is interesting and useful. The BASIC part of the routine is in lines 1930 to 2010, and the machine language routine is loaded into integer array P. It is done in the same manner as the sort routine. Although Copykill must access only one location on the disk (track 17, sector 0, MOD DO, eight bytes), I deliberately added the necessary code to the GOSUB statements (lines 1740 and 1900) to select track and sector. This is to illustrate how it is done. RW is zero for reading and one for writing.

This is a skeletal routine and the only error recovery is to display the error number on the screen. The error codes can be found in the DOS manual, pages 6-12.

The DOS Command function simply does a CMD whatever you enter and the Print Formfeed does just that. These are all the functions I've installed, the rest is up to you!

Copykill is available on cassette from the author.■

✓See List of Advertisers on page 306

```
     ELSE174Ø
1920 '          ***      DISK IO       ***
193Ø P(1Ø)=TR*1Ø+SE
194Ø IF RW=Ø THEN P(14)=17462 ELSE P(14)=17468
195Ø IF RW=Ø OR Q<>6 THEN 197Ø
196Ø IF PEEK(18151)=168 THEN POKE 18151,169 ELSE Q=1 GO
     TO 2ØØØ
197Ø P(22)=2Ø48+PS
198Ø DEFUSR=VARPTR(P(Ø)):P(2)=VARPTR(P(R1-16)):P(4)=VAR
     PTR(P(R1)):Q=USR(Z)
199Ø IF PEEK(18151)=169 THEN POKE 18151,168
2ØØØ IF Q<>Ø AND Q<>6 THEN PRINT"DISK ERROR #";Q:INPUT
     C
2Ø1Ø RETURN
2Ø2Ø '
2Ø3Ø '            ***     DOS COMMAND      ***
2Ø4Ø CLS:LINE INPUT"WHAT COMMAND ? ";A
2Ø5Ø CMD A$
2Ø6Ø IF INKEY$="" THEN 2Ø6Ø
2Ø7Ø GOTO 5ØØ
2Ø8Ø '
2Ø9Ø '           ***     LPRINT CHR$(12)      ***
21ØØ LPRINT CHR$(12);
211Ø GOTO 5ØØ
212Ø '
213Ø '          ***       READ DIR FROM SCREEN     ***
214Ø A="DIR "+PS$
215Ø IF I$="Y" THEN A=A+" (I)"
216Ø CMD A$
217Ø F="FILE DIRECTORY  ---   "+H(36)
218Ø FOR X=Ø TO 35
219Ø   Z=INSTR(H(X)," ")-1
22ØØ   F(X)=LEFT$(H(X),Z)
221Ø   IF Z=Ø THEN U=X-1:X=35
222Ø NEXT X
223Ø IF R<>3 THEN 226Ø
224Ø CLS:CMD"FREE"
225Ø F=F+"        "+H(37+PS)+" GRANS FREE"
226Ø RETURN
227Ø '          ***      INPUT SOURCE DRIVE       ***
228Ø IF QD=Ø THEN PS=Ø:GOTO 23ØØ
229Ø INPUT"SOURCE DRIVE";PS:IF PS<Ø OR PS>QD THEN 229Ø
23ØØ PS$=":"+FNA(PS)
231Ø IF R<>3 AND R<>4 THEN I$="Y":GOTO 233Ø
232Ø INPUT"INCLUDE INVISIBLE ENTRIES  (Y/N)";I$
233Ø RETURN
```

A disk rotational speed display program.

Soft Tach

Daniel North
1150 W. 29th St.
Apt. 2
Los Angeles, CA 90007

After a few unexplainable errors while using a disk system, you may begin to wonder if all that expensive hardware is shooting craps. Of course, not everyone has the equipment or the desire to electronically test and set all the operating parameters of a disk system, but there is at least one thing that can be done easily with software. Only a short BASIC program is needed to test and set the disk rotational speed.

Since all operations carried out by the TRS-80 are timed precisely by a crystal clock, the execution time for any series of operations can be calculated exactly. The program presented here uses a machine language program which counts. It starts when the index hole in the disk is sensed and stops when it passes it again. The loop which tests for a hole and increments the counter takes 16.911 microseconds with the TRS-80 clock. A disk operating at 300 RPM makes one revolution in 0.2 seconds. By dividing, it can be seen that the timing can be checked to one part in 11827 or .025 RPM. Running the program shows that disk speed will vary more than this between consecutive revolutions so it is plenty accurate.

Program Listing 1. Software Check of Disk Speed

```
10 M=32100
15 REM      POKE THE MACHINE PROGRAM INTO MEMORY
20 FORN=1TO67
30 READ D
40 POKE M,D
50 M=M+1
60 NEXTN
70 DATA 243,62,1,50,225,55,33,236,55,54,3,1,0,0,205,96
80 DATA 0,205,160,125,62,1,50,225,55,1,0,125,17,0,0,203
90 DATA 78,19,40,251,123,2,3,122,2,3,62,24,185,40,12
100 DATA 197,1,208,7,11,120,177,32,251,193,24,225,201,2
    03
110 DATA 70,32,252,201,227,227,201
120 PRINT "MEMORY SIZE MUST BE SET TO 32000"
130 INPUT "DRIVE NO. (0-3)";DN
140 DN=2[DN
145 REM      POKE THE DRIVE NUMBER INTO THE MACHINE PROG
    RAM
150 POKE 32121,DN :POKE 32102,DN
160 DEFUSR0 = 32100
170 INPUT "PRESS ENTER TO BEGIN";XY$
180 A=USR0(0)
190 REM      FIRST TWO COUNTS ARE NOT RELIABLE
200 MB=32004
210 RT=0
220 REM      GET COUNT VALUES FROM MEMORY BUFFER
230 FOR N=1TO10
240   LB=PEEK(MB)
250   MB=MB+1
260   HB=PEEK(MB)
270   MB=MB+1
280   REM      COUNTER = LOW BYTE + (HIGH BYTE * 256)
290   CO=LB+(HB*256)
300   REM      TIME PER REVOLUTION AT 4.0 MHZ
310   TM=(13022.5 + CO*7.5)/1000000
320   REM      TIME PER REV. AT 1.774 MHZ
330   TM = TM * 4/1.774
340   RPM = 60/TM
350   REM      SUM UP RPM'S TO GET AN AVERAGE
360   RT =RT+RPM
370   PRINT RPM
380 NEXTN
390 AV =RT/10
400 PRINT "AVE=";AV
410 GOTO 160
```

The program consists of two parts: one in machine language and the other in BASIC. The machine language portion is accessed from BASIC by the USR0 statement. It turns on the disk drive, waits one second for the speed to stabilize, and then counts for ten consecutive revolutions of the disk. The ten counts are stored in a buffer area set aside in memory. The BASIC part of the program PEEKs into the locations where the machine language program stored the counts. Then BASIC is used to convert the counts into 10 RPM values. These values are averaged. The 10 RPM val-

ues and the average are printed on the screen.

Since the machine language portion of this program is so short I have simply POKEd it into memory from a series of data statements. However, the assembly listing from which the data statements were generated is included.

To assemble programs which use disk it is necessary to get hold of the data sheet for the Western Digital FD1771 (the disk controller chip) or preferably the Radio Shack Expansion Interface Service Manual. The manual contains the Western Digital data sheet and other useful information. The FD1771 floppy disk controller is a complex IC with an instruction set comparable to that of the Z-80. Without knowing how the instructions work it is impossible to make sense out of any assembly listing which uses the disk.

There are two parts to the timing loop. The first part is the quick loop which looks for the hole and increments the counter. At the speed of the counter that little hole in the disk represents quite a gap. Therefore, after the hole has been sensed there is a test to see if this is the eleventh revolution. After that a loop kills time for about one tenth of a revolution to ensure that the hold has been completely passed. This second part of the loop takes a constant amount of time that must be added to the time taken by the counter. Execution of this part at 4 MHz would take 13022.5 microseconds and is the number used in the BASIC program. The reason for using the 4 MHz value is explained below.

To determine how much time is taken by a Z-80 instruction it is necessary to use the RS Editor/Assembler Manual. With each instruction in the manual there is a line that says "4 MHz E.T.:" followed by a number. The number is the amount of time needed-

Program Listing 2. Assembly Language Listing

```
7D00                 BSTRT   EQU     32000
7D64                         ORG     32100
7D64  F3                     DI
7D65  3E01                   LD      A,01H   ;DRIVE NO. POKED HERE FROM BASIC
7D67  32E137                 LD      (37E1H),A   ;FLOPPY ON
7D6A  21EC37                 LD      HL,37ECH   ;COMMAND/STATUS REG OF DISK CONTROLER
7D6D  3603                   LD      (HL),03H   ;SEEK TRACK 00
7D6F  010000                 LD      BC,00H   ;LOAD DELAY COUNTER
7D72  CD6000                 CALL    060H   ;1 SEC DELAY
7D75  CDA07D                 CALL    BUSY   ;WAIT FOR SEEK TO FINISH
7D78  3E01                   LD      A,01H   ;DRIVE NO. POKED FROM BASIC
7D7A  32E137                 LD      (37E1H),A   ;CONTINUE FLOPPY
7D7D  01007D                 LD      BC,BSTRT
                     ;THE FOLLOWING ENTRIES ARE USED FOR TIMING.
                     ;THE 4MHZ EXECUTION TIME FOR EACH INSTRUCTION IS
                     ;CIVEN IN THE TRS-80 EDITOR/ASSEMBLER MANUAL.
                     ;BY ADDING UP THE 4MHZ INSTRUCTION T'MES AND
                     ;MULTIPLYING THE RESULT BY 4/1.774 THE EXECUTION
                     ;TIME AT TRS-80 SPEED CAN BE CALCULATED.
7D80  110000         LP1     LD      DE,0000   ;DE CONTAINS TIMING COUNTER
7D83  CB4E           LP2     BIT     01,(HL)   ;TEST FOR INDEX HOLE
7D85  13                     INC     DE
7D86  28FB                   JR      Z,LP2   ;COUNT UNTIL NEXT INDEX HOLE
7D88  7B                     LD      A,E
7D89  02                     LD      (BC),A   ;LOW BYTE OF DE TO BUFFER
7D8A  03                     INC     BC   ;BUMP BUFFER POINTER
7D8B  7A                     LD      A,D
7D8C  02                     LD      (BC),A   ;HIGH BYTE TO NEXT BUFFER ADDRESS
7D8D  03                     INC     BC   ;BUMP BUFFER POINTER ACAIN
7D8E  3E18                   LD      A,24   ;NEED A TOTAL OF 12 COUNTS BECAUSE
7D90  B9                     CP      C   ;FIRST TWO MAY BE INCORRECT
7D91  280C                   JR      Z,DONE   ;LOOP FOR 12 REVOLUTIONS
7D93  C5                     PUSH    BC
7D94  01D007                 LD      BC,2000
7D97  0B             LP3     DEC     BC   ;LOOP FOR .029 SECS. APPROX. 1/10 REV.
                                     ;TO MAKE SURE INDEX HOLE HAS PASSED
7D98  78                     LD      A,B
7D99  B1                     OR      C
7D9A  20FB                   JR      NZ,LP3
7D9C  C1                     POP     BC
7D9D  18E1                   JR      LP1
7D9F  C9             DONE    RET   ;  RETURN TO THE BASIC PROGRAM
7DA0  CB46           BUSY    BIT     00,(HL)   ;(HL) CONTAINS CONROLER STATUS REGISTER
7DA2  20FC                   JR      NZ,BUSY
7DA4  C9                     RET
0000                         END
00000 TOTAL ERRORS
```


See List of Advertisers on page 306

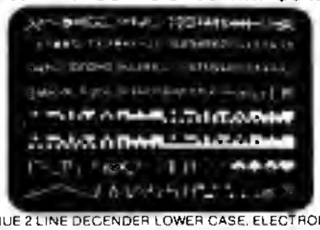
ed to execute that instruction using a 4 MHz clock. The TRS-80 uses a clock of 1.774 MHz. It would be possible to convert the times for each instruction by multiplying by 4/1.774 or about 2.2548. However, it is easier to add up all the times as if the clock were 4 MHz and then multiply by the conversion factor. For instance, the timing loop that the program uses is:

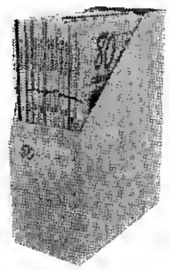
```
LP2  BIT  01,(HL)  ; time 3.00
     INC  DE       ; time 1.50
     JR   z,LP2    ; time 3.00 or 1.75
```

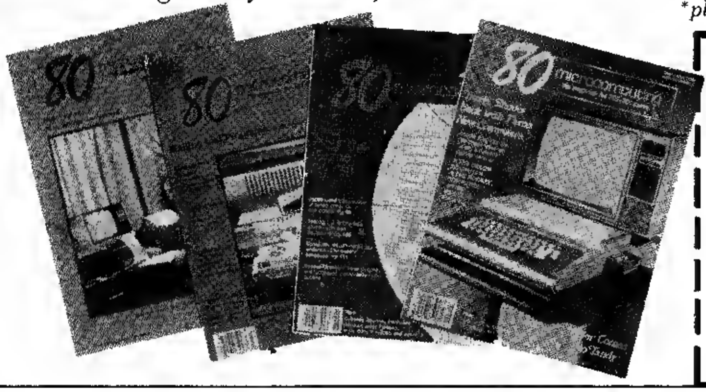
There are two times given for the last instruction because it consumes 3.00 microseconds when it loops back to LP2 and only 1.75 when it falls through. One complete loop at 4 MHz will take 7.50 μs and on the TRS-80 it will take 7.50 * (4/1.774) or 16.911. The other times are calculated in the same manner.

Examination of the assembly listing shows that the program actually uses 12 revolutions of the disk rather than 10. The count shown for the first revolution will almost certainly be in error because the first count gives only the amount of time between when the program started and when it first encountered the hole. The second count will be in error if; when the program was started, the hole just happened to be passing the sensor. These two bogus values are saved but ignored by the BASIC program.

Running the program is straightforward. Memory size is set to 32000 and the program is run. It will repeat each time Enter is pressed. I have an MPI drive, and adjusting the speed is a simple matter of removing the cover and adjusting resistor R38. The manufacturer claims accuracy to 1.5 percent so I assume that any speed between 295.5 and 304.5 is acceptable. If you have another type of drive it will be necessary to consult the manufacturer's literature to determine the allowable speed range and adjustment procedure. ■







✓See List of Advertisers on page 306

Let your 80 help you stay on top of your gasoline usage trends.

The Mileage Manager

A. L. Frink
816 S. E. 1st Court
Crystal River, FL 32629

Back when gasoline was around 60 cents a gallon, I convinced my wife Peggy that our household needed a TRS-80, if for no other reason, for the kids to enjoy. It wasn't long before she found out who the real kid was. At any rate, I went through the Level I Manual in record time and I just had to have a Level II. The Level II Manual was a little more challenging and about the time I felt comfortable with it, gas prices were climbing so I began spending my evenings writing a program to keep tabs on our gas expenses.

At that time we owned two cars: a full size 1973 Ford station wagon loaded with a 400 cubic inch engine and a 1966 Mustang with a modest 302. Even though the station wagon was a gas hog it was used more often because of its air conditioning, comfort, automatic transmission, etc.

As we started to accumulate data, it was quite a jolt to see the difference in operating costs between our two cars. As gas prices went up the differential also went up.

Not only does the program calculate miles per gallon each time the tank is filled but it calculates a summary of total miles driven, total gallons, total cost, average miles per gallon and average cost per mile. With this information available it was quite easy to compare the performance of the two vehicles.

We started sacrificing comfort for cost. (Stuffing the kids into the two-door Mustang was never popular with them, but what Dad says goes.) I was sure we were saving money, so I set

	Wagon	Mustang	Total
1979 Monthly Mileage	773	642	1415
1980 Monthly Mileage	387	839	1226
1979 Cost Per Month	$ 61	$ 32	$ 93
1980 Cost Per Month	$ 46	$ 50	$ 96
1980 Operating Cost Using 1979 Driving Habits			$ 131
1980 Operating Cost			96
		Monthly Saving	$ 35

Fig. 1. Analysis of Average Monthly Operating Cost

Program Listing 1

```
10 DEFINTA,K,N,R,X,Y,Z
20 CLEAR5000:DIMD$(30),O(30),G(30),C(30),I$(30),MP(30),
    S$(15)
30 CLS:N=1:C$="*  *":E$="$$###.##"
40 ONERRORGOTO4500
70 PRINT:PRINTTAB(8)"THE MILEAGE MANAGER"
75 PRINT:PRINTTAB(6)"* * DO YOU WANT TO * *"
80 PRINTTAB(10)"--------------"
90 PRINT:PRINT"(1) LOAD IN AN EXISTING DATA TAPE"
100 PRINT"(2) ADD NEW INFORMATION TO THE EXISTING FILE
110 PRINT"(3) PRINT A TABLE OF INFORMATION"
120 PRINT"(4) DRAW A GRAPH OF MPG VS TRIPS"
130 PRINT"(5) PRINT A SUMATION OF ALL DATA"
140 PRINT"(6) CORRECT ANY INFORMATION ON FILE"
150 PRINT"(7) SAVE THE UPDATED DATA FILE"
160 PRINT"(8) ESTABLISH YOUR FIRST DATA FILE"
170 PRINT:INPUT"ENTER A NUMBER FROM 1 TO 8";R
180 ONRGOTO500,1000,1500,2000,2400,3000,3500,4000
500 'LOAD DATA TAPE
510 IFN<>1THENCLS:GOTO70ELSECLS:PRINT"PREPARE THE RECOR
     DER TO LOAD THE DATA TAPE"
520 PRINT:GOSUB8000
540 INPUT#-1,O(0),M$
550 CLS:PRINT:PRINT
560 PRINT"THE INFORMATION IS BEING ENTERED FOR ";M$;" O
     N":PRINT
570 INPUT#-1,D$(N),O(N),G(N),C(N),I$(N)
580 PRINTTAB(20)D$(N):N=N+1:GOTO570
610 PRINT"THE DATA HAS BEEN ENTERED"
620 FORZ=1TO800:NEXT:CLS:GOTO70
1000 'ADD TO EXISTING FILE
1010 IF N=1 GOTO 4600
1020 N=N+1:GT=0:CT=0:CLS
1040 PRINT:PRINTTAB(10)C$:PRINT
1050 INPUT"ENTER THE DATE (NO COMMAS)";D$(N)
1060 PRINTTAB(10)C$
1070 INPUT"ENTER THE COST OF THE TANKFULL (NO $ SIGN)";
     C(N)
1080 PRINTTAB(10)C$
1090 INPUT"ENTER THE NUMBER OF GALLONS TO FILL THE TANK
     ";G(N)
1100 PRINTTAB(10)C$
1110 INPUT"ENTER THE ODOMETER READING";O(N)
1120 PRINTTAB(10)C$
1130 PRINT"ENTER ANY PERTINANT INFORMATION SUCH AS DRIV
     ING CONDITIONS,"
1140 PRINT"GRADE OF GAS, SPECIAL TRIPS, ETC. IF NONE PR
     ESS ENTER."
1150 I$(N)="":INPUTI$(N)
1160 IFI$(N)=""I$(N)=C$
1240 CLS:GOTO70
1500 'TABLE OF DATA
1510 IFN=1GOTO4600
1520 GOSUB5000:CLS
1530 PRINTTAB(17)"*  *  ";M$;"  *  *":PRINT
1535 IFPR$="Y"PE=10:GOSUB7000:LPRINTTAB(10)"* *  ";:LPR
     INTCHR$(27);CHR$(14);M$;CHR$(27);CHR$(15);" * *":L
```

Program continues

```
      PRINT" ":GOSUB4820
1540  GOSUB4800:Z=1:'HEADING
1560  FORX=1TON
1565  IFPR$="Y"PE=6:GOSUB7000
1570  IFZ=7Z=1:GOSUB8000:CLS:GOSUB4800
1580  IFD$(X)=C$1630
1585  IFPR$="Y"GOSUB4840
1590  PRINTTAB(0)D$(X);TAB(10)G(X);TAB(20)O(X);TAB(29)"
      ";:PRINT USING E$;C(X);:PRINTTAB(40)INT(C(X)/G(X)*
      1000)/1000;TAB(55)INT((O(X)-O(X-1))/G(X)*100)/100
1600  PRINT"--- ";I$(X);" ---"
1610  Z=Z+1:NEXTX
1625  IFPR$="Y"GOSUB4950
1630  GOSUB8000:CLS:GOTO70
2000  IFN=1GOTO4600:'GRAPH ROUTINE
2010  GOSUB5000:CLS:K=902:A=1
2020  PRINT@0,"DATE":PRINT@20,"* * M P G VS TRIPS * *"
2030  FORX=1TO57:PRINT@K+X,".";:NEXTX:K=967:'HORIZONTAL
      LINE
2060  FORY=0TO13:PRINT@Y*64+7,".";:NEXTY:'VERTICAL LINE
2100  FORX=1TON:MP(X)=INT((O(X)-O(X-1))/G(X)*100)/100:IF
      MP(X)>26A=2
2110  NEXTX
2120  FORX=0TO50STEP10:PRINT@K+X,X/2*A;:NEXTX'HORIZONTAL
       NUMBERS
2140  PRINT@K+25,"MPG";
2150  FORM=NTO1STEP-1
2160  IF(N-M)>10PRINT@140,"ONLY LAST 11 TRIPS SHOWN ";:G
      OTO2315
2170  D$="":B$="":A$="":'ABREV DATE
2180  L=LEN(D$(M)):FORY=1TO3:A$=MID$(D$(M),Y,1)
2210  B$=B$+A$:IFASC(A$)=32THEN2240
2230  NEXTY
2240  FORY=3TOL:A$=MID$(D$(M),Y,1)
2260  IFASC(A$)>47ANDASC(A$)<58ORASC(A$)=32THENB$=B$+A$
2270  NEXTY
2290  D$=LEFT$(B$,6): PRINT@832+64*(M-N),D$;
2300  PRINT@905+64*(M-N-1),STRING$(2*MP(M)/A,"=");:'PRIN
      T LINE
2310  NEXT M
2315  IFPR$="Y"PE=18:GOSUB7000:GOSUB6000
2320  A$=INKEY$:IFA$=""THEN2320ELSECLS:GOTO70
2400  IFN=1GOTO4600ELSEMT=O(N)-O(0):GT=0:CT=0:'SUMMATION
2420  FORX=1TON:GT=GT+G(X):CT=CT+C(X):NEXT
2430  MG=INT(MT/GT*1000)/1000
2450  CG=INT(CT/GT*1000)/1000
2510  GOSUB5000:CLS:PRINTTAB(11)"*  *  *  ";M$;"   *   *
        *"
2520  PRINT:PRINT"MILES DRIVEN SINCE INITIAL FILL UP *
      ";MT
2530  PRINT:PRINT"TOTAL GALLONS OF GAS SINCE ";D$(1);" *
       ";GT
2540  PRINT:PRINT"TOTAL GAS COST SINCE ";D$(1);" * ";:PR
      INT USING E$;CT
2550  PRINT:PRINT"AVERAGE MPG SINCE INITIAL FILL * ";MG
2560  PRINT:PRINT"THE AVERAGE COST PER MILE HAS BEEN  *"
      ;:PRINT USING E$;INT(CT/MT*1000)/1000
2570  PRINT:PRINT"THE AVERAGE COST PER GALLON HAS BEEN
      *";:PRINT USING E$;CG
2575  IFPR$="Y"PE=15:GOSUB7000:GOSUB4870
2580  PRINT:GOSUB8000:CLS:GOTO70
3000  IFN=1GOTO4600:'CORRECT DATA
3020  CLS:INPUT"ENTER THE DATE OF THE FILE YOU WANT TO C
      HANGE";A$
3040  FORX=1TON:IFA$=D$(X)THEN3080
3050  NEXTX
3060  CLS:PRINT"REENTER THE DATE EXACTLY AS IT IS NOW IN
       THE FILE"
3070  INPUT A$:GOTO3040
3080  CLS:PRINT"THIS IS THE DATA NOW IN THE FILE"
3090  PRINT:PRINT:GOSUB3340:PRINT
3120  INPUT"DO YOU WANT TO CHANGE THE DATE (Y/N)";A$
3130  IFA$="N"3150
3140  INPUT"WHAT DO YOU WANT THE NEW DATE TO BE";D$(X)
3150  INPUT"DO YOU WANT TO CHANGE THE ODOMETER READING (
      Y/N)";A$
3160  IFA$="N"3180
3170  INPUT"WHAT SHOULD THE ODOMETER READING BE";O(X)
3180  INPUT"DO YOU WANT TO CHANGE THE COST (Y/N)";A$
3190  IFA$="N"3210
3200  INPUT"WHAT SHOULD THE COST BE";C(X)
3210  INPUT"DO YOU WANT TO CHANGE THE NUMBER OF GALLONS
      (Y/N)";A$
3220  IFA$="N"3240
3230  INPUT"WHAT SHOULD THE GALLONS BE";G(X)
3240  INPUT"DO YOU WANT TO CHANGE OR ADD ANY PERTINENT I
      NFORMATION (Y/N)";A$
3250  IFA$="N"3280
3260  PRINT"WHAT INFORMATION DO YOU WANT IN THE FILE?"
3270  INPUTI$(X)
3280  CLS:PRINT:PRINT"THE FILE NOW READS"
3290  PRINT:PRINT:GOSUB3340
3300  GOSUB8000:CLS:GOTO70
3340  PRINTTAB(0)"DATE";TAB(12)"GALLONS";TAB(24)"ODOMETE
      R";TAB(36)"COST"
3350  PRINT
3360  PRINTTAB(0)D$(X);TAB(12)G(X);TAB(24)O(X);TAB(36)C(
      X)
3370  PRINT:PRINTI$(X):RETURN
3500  CLS:IFN=1GOTO70:'NEW DATA
```

Program continues

✓See List of Advertisers on page 306

```
3530 PRINT"SET UP A NEW DATA TAPE AND PREPARE THE RECOR
     DER"
3560 PRINT:GOSUB8000:PRINT#-1,O(0),M$
3580 FORX=1TON:PRINT#-1,D$(X),O(X),G(X),C(X),I$(X):NEXT
     X
3600 PRINT#-1,C$
3610 CLS:PRINT"A COMPLETELY NEW DATA TAPE HAS BEEN ESTA
     BLISHED"
3620 PRINT:PRINT"REWIND IT IN PREPARATION FOR THE NEXT
     ENTRY"
3630 INPUT"DO YOU WANT TO SEE THE MENU AGAIN";A$
3640 IF LEFT$(A$,1)="Y" THEN CLS:GOTO 70
3650 END
4000 'INITIAL DATA
4010 IFN<>1CLS:GOTO70
4020 CLS:PRINT"YOU SHOULD HAVE FILLED UP THE TANK TWICE
      SINCE"
4030 PRINT"TAKING AN ORIGINAL ODOMETER READING WITH A F
     ULL TANK."
4040 PRINT"IF YOU KEPT A RECORD OF MILAGE, GALLONS AND
     COST THEN"
4050 PRINT"PREPARE THE RECORDER WITH A NEW DATA TAPE"
4070 GOSUB8000:CLS
4090 INPUT"WHAT CAR IS THIS FILE FOR";M$
4110 INPUT"WHAT IS THE STARTING ODOMETER READING";O(0)
4120 PRINT#-1,O(0),M$
4130 INPUT"DATE THE TANK WAS FILLED UP";D$
4140 INPUT"ODOMETER READING WHEN WHEN FILLED";O
4150 INPUT"WHAT WAS THE NUMBER OF GALLONS";G
4160 INPUT"WHAT WAS THE COST OF THE TANKFUL";C
4170 PRINT"ENTER ANY PERTINANT OR INTERESTING INFORMATI
     ON SUCH AS"
4180 PRINT"DRIVING CONDITIONS, GRADE OF GAS, SPECIAL TR
     IPS, ETC."
4190 PRINT"IF NONE, JUST PRESS ENTER."
4200 I$="":INPUTI$:IFI$=""THENI$=C$
4220 PRINT#-1,D$,O,G,C,I$
4230 INPUT"DO YOU HAVE ANY MORE INFORMATION FOR THE DAT
     A FILE";A$
4240 IFLEFT$(A$,1)="Y"GOTO4130
4260 PRINT#-1,C$:CLS
4280 PRINT:PRINT"IN THAT CASE REWIND THE DATA TAPE. YOU
      MAY RUN THE PROGRAM AND"
4290 PRINT"ENTER THE DATA TAPE OR WAIT UNTIL YOU FILL U
     P THE TANK AGAIN."
4300 END
4500 IFERL=570N=N-1:GOTO 610:'SENSES END OF DATA
4510 PRINTERR/2+1;" ERROR ON LINE ";ERL:STOP:
4600 CLS:PRINT:PRINT"FIRST YOU MUST LOAD THE DATA TAPE"
     :GOTO520
4800 PRINTTAB(0)"DATE";TAB(10)"GALLONS";TAB(20)"ODOMETE
     R";TAB(33)"COST";TAB(40)"COST/GAL";TAB(55)"MPG":RE
     TURN
4820 LPRINTTAB(2)"DATE";TAB(10)"GALLONS";TAB(20)"ODOMET
     ER";TAB(33)"COST";TAB(42)"COST/GAL";TAB(55)"MPG";T
     AB(62)"MILES"
4825 LPRINTTAB(2)"----";TAB(10)"-------";TAB(20)"------
     --";TAB(33)"----";TAB(42)"--------";TAB(55)"---";T
     AB(62)"-----"
4826 LPRINTCHR$(138):RETURN
4840 LPRINTTAB(0)D$(X);TAB(10)G(X);TAB(20)O(X);TAB(29)"
      ";:LPRINT USING E$;C(X);:LPRINTTAB(43)INT(C(X)/G(
     X)*1000)/1000;TAB(53)INT((O(X)-O(X-1))/G(X)*100)/1
     00;TAB(62)O(X)-O(X-1)
4850 LPRINT"---";I$(X);"---":LPRINT CHR$(138):RETURN
4870  LPRINTTAB(8)"SUMMATION OF OPERATING DATA FOR"
4880 LPRINTCHR$(138);CHR$(138):LPRINTTAB(3)"* * ";:LPRI
     NTCHR$(27);CHR$(14);M$;CHR$(27);CHR$(15);"* *"
4890 LPRINTCHR$(138):LPRINT"MILES DRIVEN SINCE INITIAL
     FILL UP * ";MT
4900 LPRINTCHR$(138):LPRINT"TOTAL GALLONS OF GAS SINCE
     ";D$(1);" * ";GT
4910 LPRINTCHR$(138):LPRINT"TOTAL GAS COST SINCE ";D$(1
     );" * ";:LPRINTUSINGE$;CT
4920 LPRINTCHR$(138):LPRINT"AVERAGE MPG SINCE INITIAL F
     ILL * ";MG
4930 LPRINTCHR$(138):LPRINT"THE AVERAGE COST PER MILE H
     AS BEEN * ";:LPRINTUSINGE$;INT(CT/MT*1000)/1000
4940 LPRINTCHR$(138):LPRINT"THE AVERAGE COST PER GALLON
      HAS BEEN *";:LPRINTUSINGE$;CG
4950 FORZ=1TO4:LPRINTCHR$(138):NEXTZ:RETURN
5000 CLS:PR$="":PRINT:INPUT"DO YOU WANT A PRINTED REPOR
     T";PR$
5010 PR$=LEFT$(PR$,1):IFPR$<>"Y"RETURN
5020 INPUT"DO YOU WANT TO POSITION THE PAPER";PE$
5025 IFLEFT$(PE$,1)="N"RETURN
5030 PRINT"PRESS THE S KEY AND THE PRINTER WILL SPACE A
     HEAD"
5040 PRINT"PRESS THE R KEY WHEN THE PRINTER IS READY"
5050 IN$=INKEY$
5060 IF IN$="S" LPRINTCHR$(138):GOTO5050
5070 IF IN$="R" GOTO 5100
5080 IF IN$="" GOTO 5050
5100 POKE16424,66:POKE16425,0:RETURN
6000 'SCREEN PRINT
6010 FORT=0TO15
6020 S$(T)=""
6030 POKE VARPTR(S$(T)),64
6040 POKE VARPTR(S$(T))+1,(T*64+15360)AND255
6050 POKE VARPTR(S$(T))+2,(T*64+15360)/256
```

Program continues

Apparat, Inc. announces the most powerful Disk Operating System for the TRS-80. It has been designed for the sophisticated user and professional programmer who demands the ultimate in disk operating systems.

New 2.0 version has all of the features of NEWDOS /80 plus many more enhancements:

- Dynamic merge and delete
- Double density support (Model I with Percom and LNW Doublers)
- Mix double and single density drives
- Routing commands to send display, printer and keyboard to other devices or to a main memory location
- Disassembler will dump source code to a disk file
- Repeat function for keyboard entry
- Copy by file for single drive systems and for system files
- Copy by file with non-system disk in Drive 0
- Expanded Directories for system disks
- Create pre-allocated files
- Clear command to reset Route, Himem, Mem and others
- Forms command and Set Com RS-232 for Model III
- Chaining files may be created from SCRIPTSIT®
- Chaining greatly expanded with new features
- Date and Time saved thru non-power on reset
- Copy and Format allow default to system date
- Page scrolling while in BASIC
- Mostly upward compatible with TRSDOS 2.3 Model I
- Limited compatibility with TRSDOS Model III
- ALL NEWDOS systems maintain Model I type directories
- Model III can read and write TRSDOS Model I disk after conversion
- Global variables
- SUPERZAP & DEBUG "Find" features
- Selected copy by file for: User Files — Updated Files — Data Files
- Keep/Erase

NEWDOS/80 version 2.0 Model I or III (when ordering, please specify model number)
. . . $149.00

✓264

Apparat, Inc.

4401 South Tamarac Parkway
Denver, CO 80237

(303) 741-1778

VISA master charge

"On-going Support for Microcomputers"

```
6060 LPRINTS$(T)
6070 NEXT,T
6080 GOSUB 4950: RETURN
7000 IF PEEK(16425) >64-PE THEN LPRINT CHR$(12):IFR=3AN
     DPE<>10GOSUB4020
7010 RETURN
8000 INPUT"PRESS ENTER WHEN YOU ARE READY TO PROCEED";A
     $:RETURN
```

out to analyze all the impressive data that I had been collecting.

Back in 1979 we were averaging 773 miles per month on the gas hog and 642 on the Mustang. Today the pendulum has swung the other way. The station wagon is only averaging 387 miles per month and the Mustang is being used about 839 miles per month (see Fig. 1).

Looking at costs, I discovered that last year we were spending about $93 per month on gas. Today it is around $96. That's no savings. But it didn't take long for me to realize that the gas costs have gone up more than 50 percent and my expenses have remained almost constant.

I calculated the gasoline cost at today's prices using last year's driving habits. This told me that I was saving $35 a month.

Besides making me feel good, I now had documentation to show Peggy. Not only were we patriotic energy savers, but our computer was saving money. After coming off my high horse, I realized that the program wasn't responsible for all the savings. I am sure that some change in our driving patterns would have evolved anyway. I am confident however, that the program had an influence on us.

Running the Program

Before running the program, your gas tank should be full and the mileage recorded. The tank should then be filled at least two more times. Mileage, total cost and gallons should be recorded each time the tank is filled. Also, note any pertinent information you want kept in the file (such as highway driving, special trips, or

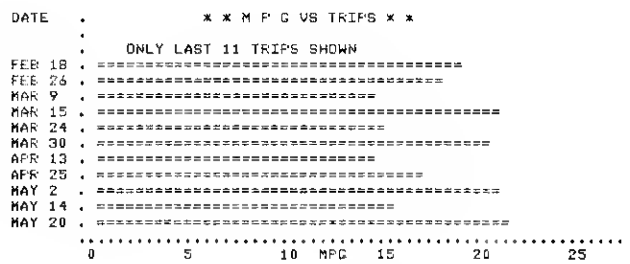
*** * * THE MUSTANG * * ***

DATE	GALLONS	ODOMETER	COST	COST/GAL	MPG	MILES
JAN 7 80	14.6	17130	$15.60	1.068	18.76	274
---* * *---						
JAN 21 80	15.1	17380	$16.50	1.092	16.55	250
---LOCAL DRIVING---						
JAN 26 80	15.2	17666	$17.30	1.138	18.81	286
---* * *---						
FEB 2 80	14.2	17909	$17.00	1.197	17.11	243
---OIL CHANGE & TUNE UP---						
FEB 9 80	10.6	18124	$12.10	1.141	20.28	215
---IN BROOKSVILLE---						
FEB 11 80	15.2	18439	$17.00	1.118	20.72	315
---TO ST PETE---						
FEB 18 80	15.7	18738	$18.00	1.146	19.04	299
---FROM ST PETE & LOCAL---						
FEB 26 80	14.5	19003	$16.70	1.151	18.27	265
---LOCAL---						
MAR 9 80	16.9	19251	$20.02	1.184	14.67	248
---LOCAL---						
MAR 15 80	18.5	19646	$21.60	1.167	21.35	395
---ST.PETE---						
MAR 24 80	11.4	19817	$13.50	1.184	15	171
---MIXED DRIVING---						
MAR 30 80	13.2	20092	$15.60	1.181	20.83	275
---ST PETE---						
APRIL 13 80	14.4	20303	$17.00	1.18	14.65	211
---LOCAL---						
APR 25 80	13.3	20535	$15.20	1.142	17.44	232
---LOCAL---						
MAY 2 80	10.5	20757	$12.00	1.142	21.14	222
---R/T TO TAMPA---						
MAY 14 80	11.1	20934	$14.00	1.261	15.94	177
---IN BROOKSVILLE---						
MAY 20 80	8.3	21114	$9.50	1.144	21.68	180
---TO DUNEDIN---						

Example 1. Average Monthly Cost Analysis

Program Listing 2

```
2010 CLS:K=900:A=0
2020 IF N=1 GOTO 4600
2030 PRINT@20,"* * M P G VS TRIPS * *"
2040 FOR X=1 TO 59
2050 PRINT@ K+X,".";:'  ** HORIZONTAL LINE **
2060 NEXT X:K=964
2070 FOR Y=0 TO 13
2080 PRINT@Y *64 +5,".";:'  **VERTICAL LINE**
2090 NEXT Y
2100 FOR X=0 TO 48 STEP 8 :'  **HORIZONTAL NUMBERS **
2110 PRINT@K+X,X/2;
2120 NEXT X
2130 FOR X=1 TO N
2140 MP(X)=INT((O(X)-O(X-1))/G(X)*100)/100:IF MP(X)>26
     THEN A=1
2150 NEXT X
2160 IF A=1 THEN GOTO  2270
2170 FOR M=0 TO 28 STEP 2:'  **VERTICAL NUMBERS **
2180 PRINT@M*64/2,28-M;
2190 NEXT M
2200 FOR Z=1 TO N:'  **IDENTIFY NUMBER OF TRIPS**
2210 IF Z>24 THEN GOTO 2380
2220 FOR Y=41 TO 43-MP(Z)*1.5 STEP-1:'  **HIGHT OF GRAP
     H**
2230 FOR X=4*Z TO 4*Z+1:'  **WIDTH OF CRAPH**
2240 SET(X+10,Y):'  ** DRAW GRAPH**
2250 NEXTX:NEXTY:NEXTZ
2260 GOTO 2370
2270 FOR M= 0 TO 56 STEP 4:'**VERT #'S UPTO 54MPG**
2280 PRINT@M*64/4,56-M;
2290 NEXT M
2300 FORZ=1 TO N
2310 IF Z>24 THEN GOTO  2380
2320 IF MP(Z)>54 THEN MP(Z)=54:'** LIMITS SIZE OF GRAPH
      TO 54MPG
2330 FOR Y= 41 TO 43-MP(Z)*.75 STEP-1:' ** HEIGHT OF GR
     APH**
2340 FOR X=4*Z TO 4*Z+1' ** WIDTH OF GRAPH **
2350 SET(X+10,Y):'  **DRAW GRAPH**
2360 NEXT X:NEXT Y:NEXTZ
2370 PRINT@1021,"";:INPUT A$:CLS:GOTO 70
2380 PRINT@140,"ONLY THE FIRST 24 TRIPS ARE SHOWN";:GOT
     O 2370
```



Example 2. Graph Report

SUMMATION OF OPERATING DATA FOR

* * * THE MUSTANG * * *

MILES DRIVEN SINCE INITIAL FILL UP * 4258

TOTAL GALLONS OF GAS SINCE JAN 7 80 * 232.7

TOTAL GAS COST SINCE JAN 7 80 * $268.62

AVERAGE MPG SINCE INITIAL FILL * 18.298

THE AVERAGE COST PER MILE HAS BEEN * $0.06

THE AVERAGE COST PER GALLON HAS BEEN * $1.15

Example 3. Option Five

M$	=	I. D. Of Car
D$(N)	=	Date
O(N)	=	Odometer
G(N)	=	Gallons
C(N)	=	Cost of Tankful
I$(N)	=	Pertinent Driving Information
MP(N)	=	Miles Per Gallon
CG(N)	=	Cost Per Gallon
GT	=	Total Gallons
CT	=	Total Cost
MT	=	Total Miles
MG	=	Average Miles Per Gallon
ML	=	Miles Driven Last Tankful
CG	=	Average Cost Per Gallon
R	=	Menu Selection
C$	=	"* *" [Default For I$(N)]
E$	=	Print Using
A$	=	Misc. Input Variables
N	=	Number Of Entries
IN$	=	Inkey
S$	=	Screen Print Routine
PR$	=	Print Request Input
K	=	Graph Variable
M	=	Graph Variable
A	=	Graph Size Variable
B$	=	Graph Date Routine
D$	=	Graph Date Routine

Table 1. List Of Variables

10- 40	Initialize
50- 180	Menu Selection
500- 620	Load Existing Data Tape
1000-1240	Add New Information To File
1500-1630	Print A Table Of Information
2000-2320	Graph Of MPG Vs Trips
2400-2580	Print A Summation Of Data
3000-3370	Correct Information In File
3500-3650	Save The Up-Dated File
4000-4300	Establish First Data File
4500-4600	Error Trap
4800-4826	Report Heading
4840-4950	Printer Report
5000-5100	Print Option And Paper Positioner
6000-6080	Screen Print Routine
7000-7010	Form Feed
8000-	Return To Menu

Table 2. Summary of Line Functions

even maintenance information). Once this information is available, the program can be run.

The program can be run on a 16K TRS-80 with or without a printer. After loading the program, a menu is displayed on the CRT with eight possible selections. Until you establish a data tape, option eight must be selected. A short introduction will give you instructions to set up the recorder with a new tape. It will ask for the name of the car and the starting odometer reading. It will next ask for information about each of the following fill-ups.

After entering the information for at least two sets of readings, the data tape can be rewound and loaded using option one. The program will ask you to load in a data tape. As the tape is being loaded, the vehicle's name and the date of each entry will be displayed on the CRT, returning to the menu when complete.

Subsequent information is added to the file using option two, which can be selected anytime after the data tape has been loaded. The program will ask for the date, the total cost of the full tank, the number of gallons, odometer and other pertinent information. When complete, the program returns to the menu.

Options three, four and five are set up for a CRT and/or a printer. If you do not have a printer, I'll explain later how to modify the program.

Option three prints a table of information. First you have the opportunity to request a printed report and to position the paper while automatically setting the line counter back to zero. A tabulation of cost per gallon, miles per gallon, miles driven, etc. will be printed (see Example 1).

Option four draws a graph of miles per gallon on specific trips. This selection draws a bar graph presenting the various dates on the Y axis and the miles per gallon on the X axis. The scale automatically adjusts dates on the matically adjusts up to either 25 MPG or 50 MPG. If there are more than 11 dates in the file, the older ones will not appear. Pressing any key will return the menu.

The graph will highlight trends in gas usage. It helped me show my 18-year-old daughter what her lead foot did to gas mileage. She wouldn't listen to me, but for some reason the computer made a believer out of her.

The graph was designed primarily for my Line Printer II, using a screen print routine. (See Example 2.) If you don't have a printer, I have included a listing of a graph routine that makes better use of the graphic capabilities of the TRS-80 CRT.

1520	CLS
1535	Delete
1565	Delete
1585	Delete
1625	Delete
2000 - 2320	Delete
and replace with Program Listing 2	
2575	Delete
4820 - 7010	Delete

Table 3

Option five will print a summation of total miles driven, total gallons of gas, total cost of gas, average MPG, the average cost per mile and the average cost per gallon. Example 3 is an example of the summary.

Any data based program should have provisions for changing data in the file. These provisions are provided by option six. By entering the date, the program will search the file and permit you to change any information in that record.

Option seven saves the updated file. In order to save new entries made using option two or corrections made in option six, a new data tape must be established.

A list of variables is included in Table 1 and a summary of the line functions in Table 2.

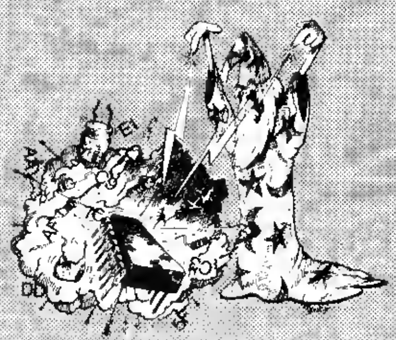
If you haven't added a printer to your system yet, the program can be used without problem, or modified by making the changes indicated in Table 3.

This program by itself won't save you any money, but it can be a fun way to manage your gas expenses and document car maintenance. And if conditions are right, it might help you save $35 a month like it does for me. ■

✓See List of Advertisers on page 306

The books you've been waiting for...

FUTURE MYSTERIES



Dennis Bathory Kitsz
The CUSTOM TRS-80

Available June

More than 300 pages of TRS-80 customizing information. With this book you will be able to explore your computer like never before. Want to turn an 8 track into a mass storage unit? Individual reverse characters? Build a high resolution graphics board? Replace the BASIC ROMs? Make music? High speed, reverse video, Level I & Level II? Fix it if it breaks down? All this and much, much more. Even if you have never used a soldering iron, or read a circuit diagram, this book will teach you how to. This is going to be the definitive guide to customizing your 80!

Lewis Rosenfelder
BASIC FASTER & BETTER

Available August

You don't have to learn assembly language to make your programs run fast. With the dozens of programing tricks and techniques in this book you can sort at high speed, swap screens in the twinkling of an eye, write INKEY routines then people think are in assembly language and add your own commands to BASIC. Find out how to write elegant code that really makes your BASIC hum, and explore the power of USR calls. Tested examples complement the explanations, and all the programs will be available on disk.

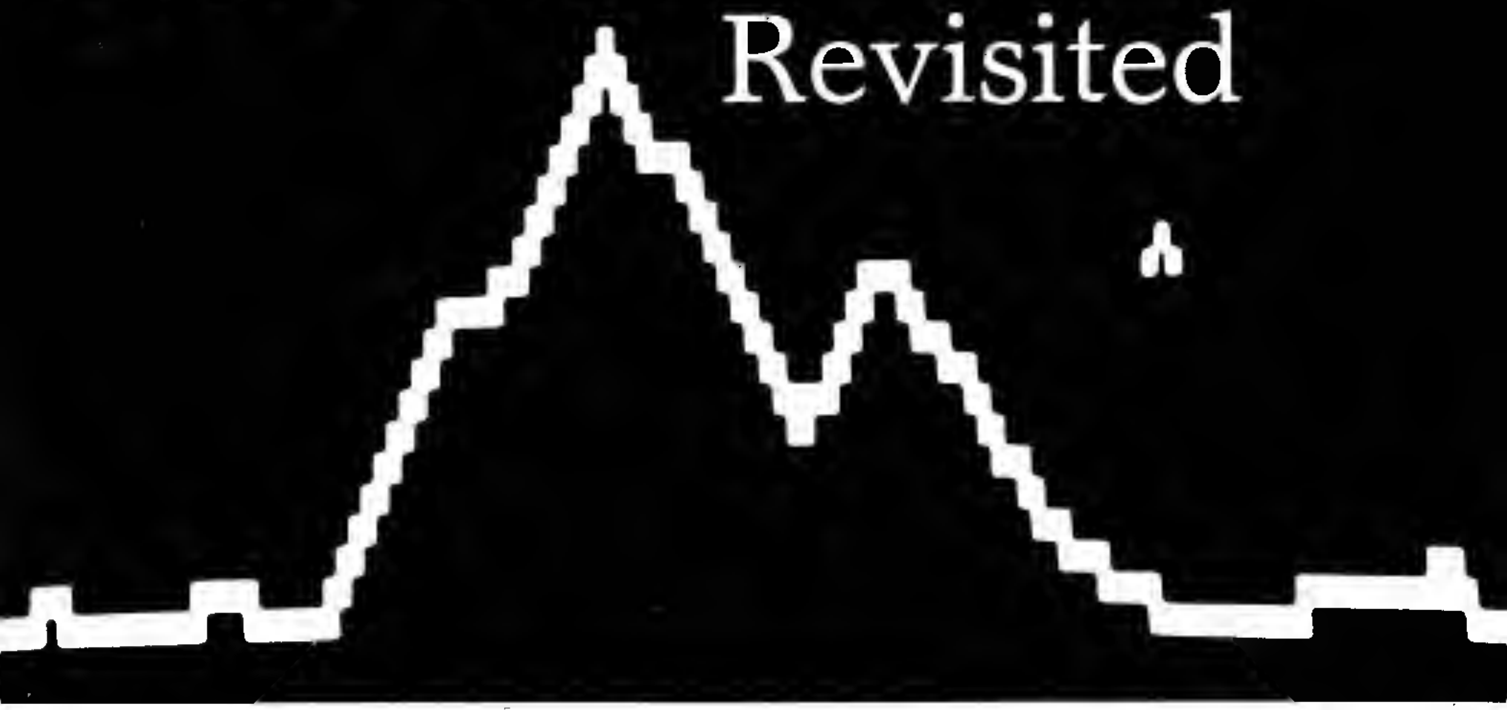
Look for them at your IJG dealer.

Get them at your local IJG dealer!

Now available at selected DALTON **bookstores**

Anderson Computers Huntsville AL
Personal Computer Place Mesa AZ
Simutek Computer Products Tucson AZ
Berkley Microcomputer Berkley CA
Thorpe Dataservices Canoga Park CA
Data Dynamics Cerritos CA
J.K. Gill Cupertino CA
P C Computers El Cerrito CA
Radio Shack Associate Store El Cajon CA
Computerland South Bay Lawndale CA
M M & T Industries Lompoc CA
E.T. Research Long Beach CA
Op Amp Technical Books Los Angeles CA
Small Systems Software Newbury Park CA
Hobby World Northridge CA
Kive Computers Oceanside CA
Stacey's Palo Alto CA
Byte Shop of Placentia Placentia CA
Capital Computer Systems Sacramento CA
Computerland San Bernadino CA
Stacey's San Bernadino CA
Computerland San Diego CA
Computer Age Inc San Diego CA
Disney's Electronics San Diego CA
Stacey's San Francisco CA
Computer Store of San Leandro San Leandro CA
Computer Information Exchange San Luis Rey CA
Microcomputer Technology Santa Ana CA
Vista Computer Co Santa Ana CA
Radio Shack Associate Store Sonora CA
Exatron Sunnyvale CA
James Games Computer Center Upland CA
Computer Land of West L.A. West L.A. CA
Computers To Go Westlake Village CA
Apparat Denver CO
Software Gourmet Denver CO
Computer Works Westport CT
The Program Store Washington DC
Computer Junction Fort Lauderdale FL
Adventure International Fern Park FL
HIS Computermation Melbourne FL
Boyd Electronics Panama City FL
TAB Sales Company Pensacola FL
AMF Microcomputer Center Tampa FL
Atlanta Computer Mart Atlanta GA
The Bottom Shelf Atlanta GA
Radio Shack Associate Store Honolulu HI
Idaho Microcomputers Buhl ID
Garcia & Associates Chicago IL
Data Domain of Schaumberg Schaumberg IL
Bob's Sales Agency London KY
Computer Shoppe Metairie LA
Tulane Book Store Tulane LA
Mark Gordon Computers Cambridge MA
Computer City Charlestown MA
Radio Shack Associate Store Falmouth MA
Radio Shack Associate Store Marlborough MA
Miller Microcomputer Natick MA
Radio Shack Associate Store New Bedford MA
Either/Or Bookstore Pittsfield MA
Small Business Systems Group Westford MA
Radio Shack Associate Store Calais ME
Radio Shack Associate Store Caribou ME
Radio Shack Associate Store East Wilton ME
Eastern Book Company Portland ME
A M Electronics Ann Arbor MI
Wizards Arsenal East Lansing MI
Computer Center Garden City MI
Soft Sector Marketing Garden City MI
The Alternate Source Lansing MI
Level IV Products Livonia MI
C & S Electronics Milan MI
The Golden Anvil South Haven MI
Software Shack Associate Store Belton MO
The Software Center Florissant MO
Computer Mart Springfield MO
Radio Shack Associate Store Lebanon NH
Radio Shack Associate Store Portsmouth NH
Computer City Manchester NH
Ramworks Milford NH
73 Inc Peterborough NH
Radio Shack Associate Store Freehold NJ
Radio Shack Associate Store Middletown NJ
Westech Sparta NJ
Industrial Products Corp Albuquerque NM
Radio Shack Associate Store Elko NV
Hurley Electronics Las Vegas NV
B T Enterprises Centereach NY
Digibyte Systems New York NY
Integrel Business Systems New York NY
Transbooks Inc. New York NY
H & E Computronics Spring Valley NY
The Computer Corner White Plains NY
Computer Resource Williamsville NY
Micromini Computer World Columbus OH
Meta Technology Euclid OH
Astro Video Electronics Lancaster OH
Vern Street Products Sapulpa OK
Computer Store Tulsa OK
Sounds Etc Watonga OK
Erie Computer Store Erie PA
VR Data Folcroft PA
Marketline Systems Huntingdon Valley PA
Computerland Mechanicsburg PA
Personal Computer Corp Paoli PA
Whitman Book Shop Philadelphia PA
Computer House of Pittsburg Pittsburg PA
Brodart Inc Williamsport PA
Computerlab Memphis TN
Computer Fort Arlington TX
Quality Software Dallas TX
Waghalter Books Houston TX
Quality Technology Salt Lake City UT
Computer Shoppe Metairie VA
Home Computer Center Virginia Beach VA
Virginia Microsystems Woodbridge VA
Claude's TV Newport VT
H & S Microsystems Blaine WA
Magnolia Microsystems Seattle WA
Personal Computers Inc Spokane WA
Byte Shop of Milwaukee Milwaukee WI
Mc Gill's Australia
Micro 80 Australia
Pitt Street Microcomputers Australia
Technical Book and Magazine Australia
Alpha 1 Micro Canada
HRB Goderich Canada
University Bookroom Canada
EC Data Denmark
A.J. Harding England
Mine of Information England
The Computer Bookshop England
Radcom Electronics Holland
Academia Boekh Netherlands
Nordisk Mikro Sweden
Hofacker GMBH West Germany
Interface Age (Europe) West Germany

Lunar Lander Revisited



"The object of the game is to skim the top of the mountains and land at the bottom."

John Beringer
2729 W. Sahara
Las Vegas, NV 89102

A game that has been debugged to death quickly loses its challenge. Not so with Lunar Lander.

This version does not wait for input. Instead, it examines memory to determine if a key has been pressed, and whether to perform the corresponding action—firing the landing rocket or the lateral jets. The object of the game is to skim the top of the mountains and land at the bottom.

Numeric keys 1 through 7 determine the burn rate of fuel for the main landing rocket. These keys may be pressed for continuous firing, or tepped for bursts of fire. The burn rate is geometric, one key producing twice as much power as the next lower key. Thus, the player burns no fuel when no key is pressed, and may vary the burn rate depending on the key pressed.

The < (,) and > (.) keys fire the lateral jets. Either may be pressed at any time, even when the main engine is firing. These two keys may also be pressed for continuous firing, but do use up fuel. Don't overshoot the base, or you'll burn up more fuel getting back.

When fire flickers from your ship's tail, the main engine is firing. The computer will also make comments on your landing.

Oh yes, the more you play, the harder the game becomes, as the computer will give you less starting fuel after a few good landings.

Happy landing! ■

Program Listing

```
900 GD=Ø:BD=Ø
950 DIM STARS(7):ST(1)=23:ST(2)=54:ST(3)=79:ST(4)=105:S
     T(5)=133:ST(6)=162:ST(7)=175
1000 CLS:PRINTCHR$(23);:PRINT@276,"REAL TIME":PRINT@400
     ,"LUNAR  LANDER":PRINT:PRINT"YOUR NAME, CAPTAIN";:
     INPUT NA$
1100 CLS
1110 PRINT@28,"A L E R T !":PRINT:PRINT"EMERGENCY, CAPT
     AIN ";NA$:PRINT:PRINT"NAVIGATIONAL COMPUTER MALFUN
     CTION":PRINT:PRINT"YOU WILL HAVE TO LAND BY THE SE
     AT OF YOUR PANTS!":PRINT"< CAUSES LEFT LATERAL MOT
     ION":PRINT"> CAUSES RIGHT LATERAL MOTION"
1115 PRINT"FUEL BURN RATE IS FROM 1 - 7; NO KEY IS ZERO
      BURN RATE":PRINT"LAND AS NEAR THE BASE AS POSSIBL
     E":PRINT:PRINT"PRESS ANY KEY TO START"
1300 FOR N=1 TO 5Ø:NEXT N:PRINT@28,"             ";:FOR N
     =1 TO 5Ø:NEXT N:PRINT@28,"A L E R T !";:IF INKEY$
     = "" GOTO 13ØØ
2100 REM BUILD TERRAIN
2110 CLS:FOR X=Ø TO 7:SET(X,44):NEXT X:FOR X=7 TO 9:SET
     (X,43):NEXT X:FOR X=9 TO 19:SET(X,44):NEXT X:FOR X
     =19 TO 23:SET(X,43):NEXT X:FOR X=23 TO 27:SET(X,44
     ):NEXT X:Y=44:FOR X=28 TO 38:SET(X,Y):SET(X+1,Y):Y
     =Y-1:NEXT X
2115 FOR X=39 TO 42:SET(X,34):NEXT X:SET(42,33):Y=33:FO
     R X=43 TO 5Ø:SET(X,Y):SET(X+1,Y):Y=Y-1:NEXT X:SET(
     51,25)
2120 Y=26:FOR X=51 TO 54:SET(X,Y):SET(X+1,Y):Y=Y+1:NEXT
      X:SET(56,29):SET(57,29):Y=29:FOR X=56 TO 65:SET(X
     ,Y):SET(X+1,Y):Y=Y+1:NEXT X:SET(65,37):SET(66,37):
     Y=37:FOR X=67 TO 71:SET(X,Y):SET(X+1,Y):Y=Y-1:NEXT
      X:Y=33:FOR X=73 TO 75:SET(X,Y):SET(X+1,Y):Y=Y+1:N
      EXTX
2125 SET(77,35):SET(77,36):Y=36:FOR X=78 TO 81:SET(X,Y)
```

Program continues

```
     :SET(X+1,Y):Y=Y+1:NEXT X:Y=39:FOR X=82 TO 84:SET(X
     ,Y):SET(X+1,Y):Y=Y+1:NEXT X:SET(86,41)
2130 FOR X=86 TO 89:SET(X,42):NEXT X:FOR X=89 TO 93:SET
     (X,43):NEXT X:FOR X=93 TO 105:SET(X,44):NEXT X:FOR
     X=105 TO 119:SET(X,43):NEXT:FOR X=116 TO 118:SET(
     X,42):NEXT:FOR X=119 TO 127:SET(X,44):NEXT
2150 FOR N=1 TO 7:PRINT@ST(N),".";:NEXT
2200 GRAV=6.4
2210 VEL=RND(50)+10
2220 TIME=.5:FUEL=700
2230 IF GD>4 FUEL=650 ELSE IF GD>9 FUEL=600
2250 PIC=0
2251 REM PIC= -1 BIT ON
2252 REM PIC=  0 BIT OFF
2253 REM PIC=  1 STAR
2560 LY=2:LY=2:X=2:ALT=430
2570 HI=0
2900 PRINT@960,"FUEL";:PRINT@975,"VELOCITY";:PRINT@995,
     "ALTITUDE";:PRINT@1017,"BASE";
3000 IF FUEL>0 THEN GOTO 3100 ELSE BURN=0:GOTO 3900
3100 BURN=PEEK(14352)/2
3900 FUEL=INT(FUEL-BURN*TIME):IF FUEL<1 THEN FUEL=0
4000 PRINT@965,FUEL;:VEL=INT(VEL-BURN*TIME+GRAV*TIME*TI
     ME)::PRINT@984,VEL;:ALT=INT(ALT-VEL*TIME):IF ALT<1
     THEN ALT=0
4001 PRINT@1004,ALT-10;:Y=43-INT(ALT/10)
4010 IF ALT>0 GOTO 4070
4011 IF Y<1 THEN Y=1
4012 IF VEL>15 GOTO 4050 ELSE IF VEL>5 GOTO 4039
4015 IF VEL<0 GOTO 4100
4019 IF Y=41 THEN Y=42
4020 RESET(LX,LY):RESET(LX+1,LY-1):RESET(LX+2,LY):SET(X
     ,Y):SET(X+1,Y-1):SET(X+2,Y)
4022 IFLX>113ANDLX<121THENGOSUB9100:PRINT@851,"NICE LAN
     DING, SPORT";:PRINT@915,"BUT YOU HIT THE BASE!";:G
     OTO 9200 ELSEIFLX>119ORLX<105THENPRINT@851,"NOT TO
     O BAD, BUT ITS";:PRINT@915,"A LONG WALK HOME!";:GO
     TO 9200
4023 G=RND(2):ON G GOSUB 4025,4026,4027:GOTO 9300
4025 PRINT@851,"RIGHT ON THE MONEY!";:RETURN
4026 PRINT@851,"PROMOTION ON THE WAY!";:RETURN
4027 PRINT@851,"YOU'RE A NATURAL";:PRINT@915,"STAR PILO
     T";:RETURN
4039 IF Y=41 THEN Y=42
4040 RESET(LX,LY):RESET(LX+1,LY-1):RESET(LX+2,LY):SET(X
     ,Y+1):SET(X+1,Y):SET(X+1,Y+1):SET(X+2,Y+1)
4041 IF X>0 SET(X-1,Y)
4042 IF X<125 SET(X+3,Y)
4043 G=RND(3):ON G GOSUB 4045,4046,4047,4048
4044 IF LX>113 AND LX<121 THEN GOSUB 9100:PRINT@746,"AN
     D YOU HIT THE BASE!!";: GOTO 9200 ELSE GOTO 9200
4045 PRINT@851,"YOU SAVED THE CARGO";:PRINT@913,"BUT YO
     U SMASHED THE CREW!";:RETURN
4046 PRINT@851,"YOU SAVED THE CREW";:PRINT@913,"BUT YOU
     SMASHED THE CARGO!";:RETURN
4047 PRINT@851,"UFF!";:PRINT@915,"SHE'LL NEVER FLY AGAI
     N!";:RETURN
4048 PRINT@851,"NOT SO GOOD,";:PRINT@915,"L I E U T E N
     A N T !!";:RETURN
4050 RESET(LX,LY):RESET(LX+1,LY-1):RESET(LX+2,LY):RESET
     (X+1,Y):Y=Y+1:SET(X,Y-1):SET(X+2,Y-1):RESET(X,Y):R
     ESET(X+2,Y):RESET(X+1,Y-1)
4051 IF X>0 SET(X-1,Y-2)
4052 IF X<125 SET(X+3,Y-2)
4053 IF POINT(X+1,Y)=0 SET(X+1,Y)
4055 PRINT@64,CHR$(30);:IFX<28ORX>89PRINT"NOTE THE NEW
     LUNAR FORMATION ...":PRINT"IT IS THE CRATER OF ";N
     A$: GOTO 9200 ELSEPRINT"COULDN'T YOU SEE THAT MOUN
     TAIN, ";NA$;"?":PRINT"WERE YOU TRYING TO DRILL A T
     UNNEL TO THE OTHER SIDE?";:GOTO 9200
4070 RESET(BX,BY):IF Y<0 GOTO 4150 ELSE IF POINT(X,Y+1)
     =-1 OR POINT(X+1,Y+1)=-1  OR POINT(X+2,Y+1)=-1 GOT
     O 9000
4100 RESET(BX,BY)
4150 IF Y<2 THEN RESET(LX,LY):RESET(LX+1,LY-1):RESET(LX
     +2,LY)
4200 IF FUEL <5 GOTO 4220 ELSE IF PEEK(14368)=64 THEN X
     =X+1:FUEL=FUEL-5 ELSE IF PEEK(14368)=16 THEN X=X-1
     :FUEL=FUEL-5
4210 IF X>124 THEN X=X-1 ELSE IF X<0 THEN X=X+1
4220 IF X=LX AND Y=LY GOTO 8999 ELSE IF Y<1 GOTO 8999
4400 RESET(LX,LY):RESET(LX+1,LY-1):RESET(LX+2,LY):IF Y<
     12 THEN FOR N=1 TO 7:PRINT@ST(N),".";:NEXT N
4405 IF PIC=-1 THEN SET(LX+1,LY+1)
4410 SET(X,Y):SET(X+1,Y-1):SET(X+2,Y):LX=X:LY=Y:BX=X+1:
     BY=Y+1
4420 PIC=POINT(BX,BY):IF BURN>1 SET(BX,BY)
8999 GOTO 3000
9000 IF ALT<230 GOTO 4011 ELSE GOTO 4150
9100 SET(113,41):SET(114,42):RESET(115,42):SET(116,42):
     RESET(117,42):SET(118,42):SET(119,41):FOR X=114 TO
     118:RESET(X,41):NEXT:RESET(119,42):RESET(120,42):
     RETURN
9200 BD=BD+1:GOTO 9400
9300 GD=GD+1
9400 PRINT@192,"GOOD LANDINGS SO FAR... ";GD:PRINT"BAD
     LANDINGS SO FAR.... ";BD:INPUT"TRY YOUR LUCK AGAIN
     ? Y OR N";R$:IF R$="Y" THEN GOTO 2100 ELSE IF R$="
     N" END ELSE GOTO 9400
9999 END
```

Hold that printout! Here's a software patch that Tandy neglected.

Underscoring Scripsit

Carl Iseli
2108 Kingshouse Road
Silver Spring, MD 20904

If you are already using Radio Shack's Scripsit word processing system, you know it's one of the best available. Its unlimited formatting capabilities within the text is probably the closest thing to guaranteeing freedom of expression since the First Amendment.

For all of Scripsit's attributes, it does have several weaknesses. The greatest of these are its inabilities to underline text (or even print the underscore character, for that matter) and to halt printing (in order to change type fonts, etc.). Fortunately, this and other print related shortcomings can be easily corrected—with no hardware modifications.

The root of the underscore and special character problem is in the hardware of the TRS-80. Specifically, the TRS-80 keyboard has neither dedicated special character keys nor programmable keys. We'll have to invent some special keys with software.

First the Bad News

In order to dedicate a special key in Scripsit, you will have to give up one that is normally used for other purposes. I chose to give up the "at" character, which is generated in Scripsit by hitting the shift key, followed by the number 0. I chose the "at" character because in any situation I could always type the word "at" instead of using the character. If you use Scripsit to type and edit your BASIC programs, you will still have the use of the "at" character; it will LIST, LLIST and run properly under BASIC, but you will have to give up the ability to print the program using Scripsit.

The other bad news is perhaps a blessing in disguise. After modifying Scripsit, you will use various combinations of the "at" character with other characters to perform text underlining and to cause printer pauses.

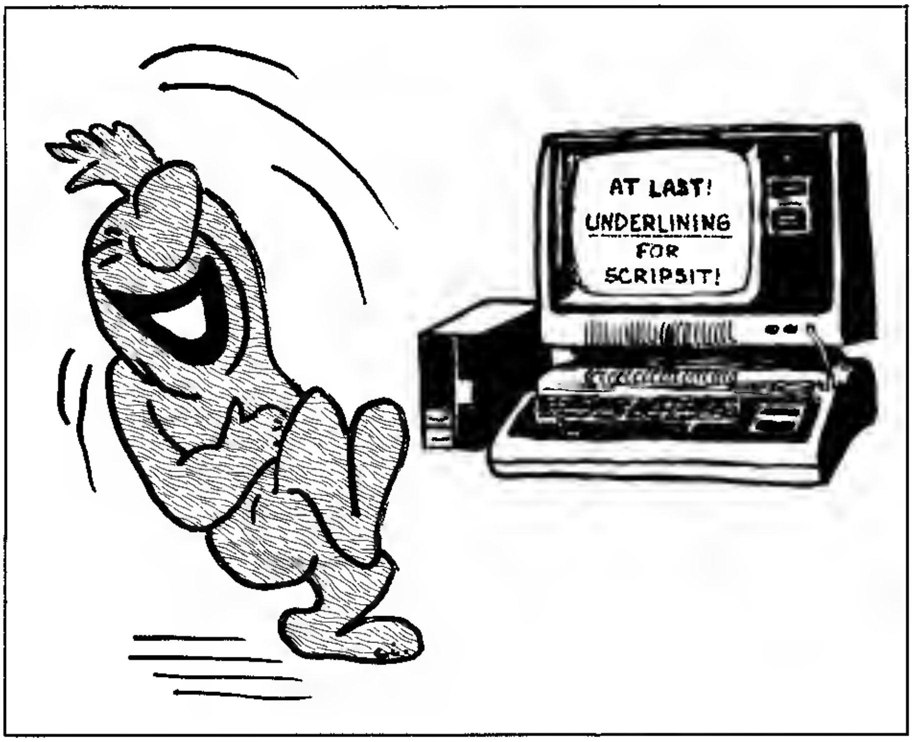
While the characters used this way will be completely invisible to the printer, the Scripsit program will still see them as printable characters. The program will, therefore, count these characters in its calculation of the line length. The consequence of this is that you will not be able to print underlined text in justified (flush right and left) format.

I say this may really be a blessing because text that is justified by inserting extra spaces between words, rather than proportionally spacing the letters within each word, generally looks strange and is difficult to read.

Now, the Good News!

Other than the changes above, Scripsit will operate normally. Once the modifications are made, you can form lines of underscore characters simply by typing a string of "at" characters. You can underline words or phrases of text by preceding the first letter of the text to be underlined with an "at" character and typing another "at" character immediately following the last letter of the text to be underlined. Finally, you can cause your printer to pause by embedding an "at" character in your text followed immediately by a greater than sign (>).

In addition to the Scripsit program itself, you will need a minimum of 32K RAM, at



"Once the modifications are made, you can form lines of underscore characters simply by typing a string of 'at' characters."

least one disk drive and a parallel printer with backspacing (e.g. Radio Shack WP50, NEC Spinwriter 5530, Diablos and Qumes). Radio Shack's Editor/Assembler, or its equivalent, is helpful in entering the machine code; but it is not entirely necessary. I have included listings that can be entered by DOS Debug for those who do not have an assembler available.

If you shrink in terror at the mention of machine code or assembly language, don't drop out: I'll go through the modification steps one by one.

Getting Started

Prepare a disk with a copy of unmodified Scripsit on it, and make sure that you have a minimum of 20 granules of free space available. You will be placing two small programs on it, plus a core image of the interim modifications and the final, fully modified Scripsit program itself.

The first step relocates Scripsit to a high RAM address. This must be done because Scripsit normally resides in RAM at addresses starting at 5200H, and thus overlays the area used for DOS commands. By relocating Scripsit, we can work on it to our heart's content, then put it back in its resident RAM position just before execution.

Using your editor/assembler, type in the relocator program given in Program Listing 1. Make a tape of the object code, then return to DOS and type TAPEDISK. When the prompt appears, type C to load the object file. When the prompt returns, type E to return to DOS.

If you don't have an editor/assembler available, you can use the Debug M command: Type DEBUG and enter, then push the Break key to enter the Debug routine. Type DBFOO (enter), then MBFOO and press the space bar. Your display should now have the hex number located at BFOOH bracketed, and the lower left portion of the screen should read BFOO.

Now type the following sequence of hex numbers, pressing the space bar after each pair of digits:

F3 21 00 52 11 00 82 01 A4 28 E5 B7 ED 52 E1 ED B0 C3 00 00

Return to DOS by entering G402D.

Whichever way you produced your relocator program, you'll want to make a disk file of it. Enter TAPEDISK, type F RELO/CIM:0 BF00 BF13 BF00 and enter it. Then return to DOS by typing E.

Load, but do not run, Scripsit by typing LOAD SCRIPSIT/LC. When the load is complete, type BASIC2 (enter). This will put you in Level II BASIC and the screen will ask for Memory Size?, to which you reply Enter.

Now type SYSTEM and execute the relocation program by answering the prompt with /48896 (enter). The relocation program will instantly zap Scripsit to high memory and return you to DOS. This is done in the wink of an eye, so don't worry that nothing could have happened before you returned to DOS—it did!

Just to make sure everything done so far has been correct, type DEBUG, enter it, then press Break. After the Debug format appears on the screen, type D8200 and enter it. If all is well, the fourth line from the bottom of your screen should now read:

8200 18 3D 43 4F 50. . .

Return to DOS by entering G402D.

Now, let's save the relocated Scripsit for future use. Type DUMP SCRIPSIT/CIM (START = X'8200', END = X'AAA3'), then press ENTER. You will now have a non-executing file that you can use for these and other Scripsit modifications.

For the next phase of the modification, you have two choices: If you own an editor/assembler, utilize the following section, "Using an Editor/Assembler." If not, skip the next section completely and follow the instructions in the section labeled "Doing it the Hard Way." O.K., let's go our separate ways; then meet again in the section titled "Everybody, Altogether Now."

Using an Editor/Assembler

If you have an editor/assembler available, use it to type in the assembly language modification exactly as it appears in Program Listing 2. If your assembler will write

FE 40	28 09	CD CD	7A 32	E8 37	C3 74	5F 08	3A 26	7B B7
28 66	7E FE	3E 28	66 FE	40 28	28 FE	21 38	24 FE	80 30
20 3E	00 32	26 7B	32 28	7B 08	C9 08	3A 26	7B B7	20 0D
08 FE	0E 38	14 08	3A 28	7B 3C	32 28	7B 08	C9 3E	FF 32
26 7B	08 3E	5F 18	B8 08	C5 0E	08 CD	01 7B	0E 5F	CD 01
7B 3E	FF 32	26 7B	C1 08	C9 3A	28 7B	47 79	32 E8	37 CD
10 7B	05 20	F6 C9	3A E8	37 CB	7F 20	F9 C9	08 CD	ED 7A
C9 23	3A 40	38 B7	28 FA	08 C9	21 E3	03 22	16 40	F3 21
00 82	11 00	52 01	27 29	ED B0	3E FF	32 26	7B C3	00 52

Table 1

4D 4F	44 20	55 4C	50 20	28 43	29 20	43 41	52 4C
20 49	53 45	4C 49	2C 20	31 39	38 30	20 20	41 4C
4C 20	52 49	47 48	54 53	20 52	45 53	45 52	56 45

Table 2

```
                00100 ;        PROGRAM 1, RELO/CIM
                00110 ;        RELOCATES SCRIPSIT FROM
                00120 ;        5200-7AA4 TO
                00130 ;        8200-AAA4
                00140 ;
                00150 ;        LOADS AT BFOOH (48896)
                00160 ;
BF00            00170          ORG      0BF00H
BF00 F3         00180          DI                      ;STOP THE CLOCK
BF01 210052     00190          LD       HL,5200H       ;START ADDRESS
BF04 110082     00200          LD       DE,8200H       ;DESTINATION
BF07 01A428     00210          LD       BC,28A4H       ;# OF BYTES TO MOVE
BF0A E5         00220          PUSH     HL             ;SAVE START ADDR
BF0B B7         00230          OR       A              ;CLEAR CARRY FLAG
BF0C ED52       00240          SBC      HL,DE          ;WHICH DIRECTION?
BF0E E1         00250          POP      HL             ;RESTORE START ADDR
BF0F EDB0       00260          LDIR                    ;MOVE IT!
BF11 C30000     00270          JP       0000H          ;BOOT THE SYSTEM
0000            00280          END
00000 TOTAL ERRORS
```

Program Listing 1

"Without an editor/assembler, you may want to rest your fingertips before going on . . ."

the object code directly to disk, do so; or save it on tape and use the DOS Tapedisk utility to put it on disk. In any case, use the filespec MODULP/CIM.

In DOS mode, enter LOAD SCRIPSIT/CIM, then enter LOAD MODULP/CIM. Now type DUMP SCRIPSIT/CIM (START = X'8200', END = X'AB40'), and enter it. That's it. You can skip the next section unless you're a masochist.

Doing It the Hard Way

Without an editor/assembler, you may want to rest your fingertips before going on—you've got some serious typing to do. It will be worth it in the long run, though.

In the DOS mode enter LOAD SCRIPSIT/CIM;. Now enter DEBUG and press Break. Your screen should soon be filled with all the register contents. When it is, type DAA00 (enter); then type S and enter it to get a full screen of characters starting at AA00H.

Now type MAA9E and press the space bar. At the bottom left of your screen you should see 32 –, and the hex number 32 at location AA9EH should be bracketed by graphics bars. If everything is O.K. so far, enter the sequence of hex numbers shown in Table 1. Be sure to press the space bar after each two-digit number is entered.

If you entered all the numbers in Table 1, the bottom left corner of your screen should now indicate the address AB40. (This doesn't prove that you entered them all correctly, just that you entered them all.)

Press X, followed by G402D (enter), and not only will you be returned to DOS but the worst part will be over! Type DUMP SCRIPSIT/CIM (START = X'8200', END = X'AB40'), and enter it. You are now ready for the final step in your modification.

Everybody, Altogether Now

There is only one last step between us and our completed Scripsit modification. We'll do it by hand-assembly just so all you editor/assembler owners will remember the good old days. In the DOS mode enter LOAD SCRIPSIT/CIM;. Now enter Debug and press Break.

Your screen should soon be filled with all the register contents. When it is, type D8800 and enter it. Now type M8800 and press the space bar. At the bottom left of your screen you should see 20 –, and the numbers 20 at location 8800H should be bracketed by graphics bars. Enter the sequence of hex numbers in Table 2, again pressing the space bar after typing each two-digit number:

The bottom left corner of your screen should now give the address AB40 followed by 13 –. Press X, followed by G402D (enter), and you will be returned to DOS. Now type DUMP SCRIPMOD/CMD (START = X'8200', END = X'AB40', TRA = X'AB26') and enter it.

If you have followed the above instructions without error, you will now have a full copy of underline and pause-modified Scripsit! You can call the program from DOS at any time by simply typing SCRIPMOD (enter).

Theory of Operation

Scripsit's actual line printer routine, as far as our interests are concerned, starts at location 7A9EH, which we have relocated to AA9EH. Here we would normally find: LD (37E8H),A, which outputs the character in the A register to the printer, and JP 5F74H, which jumps to a part of Scripsit to perform all sorts of housekeeping and get the next printable character.

Fortunately for our purposes, these locations coincide with the end of Scripsit, and are followed by a relatively large unused area of RAM (do the Radio Shack programmers have some future plans up their collective sleeves?). It is here that we interrupted the normal flow of Scripsit, inserted our modification, then plugged back into the program.

Referring to the assembly language Program Listing 2, you may find the address areas of the CALLs and DEFLs a bit unusual. They were coded this way to save time during the many program revisions I had to make.

This form of addressing makes the code totally relocatable. When changing the ORG statement, or whenever program material is inserted or deleted, all the calls are automatically updated. I could have replaced them with the actual hex addresses of the final program, but I thought it was a neat trick worth passing along.

Lines 140–150: FLAG and CNTR are assigned to indicate the RAM locations to be used for a flag and a counter.

Lines 170–180: The next character about to be sent to the printer is checked to see if it is an "at" character (40H). If it is, control is transferred to line 220.

Line 190 calls a subroutine at line 400 that checks the flag status.

Line 200 sends the character in the A register to the printer.

Line 210 returns program flow to the main body of Scripsit.

Lines 220–390: Here, the flag status is checked. If it is set (00 in location 7B26H), control is passed to the underline routine beginning at line 810. If not, the routine looks at the next text character. If the next character is a greater than symbol, program flow reverts to the printer pause routine at lines 840–890. If it is another "at" symbol, a space (coded or uncoded), a linefeed or a carriage return, the original char-

"While this may seem like a long and complicated routine, you won't actually notice any slow-up in your printing. This is because any word processing printer is almost infinitely slower in accepting characters. . ."

acter is converted to an underscore (5FH) and printed, using the routine at lines 530–570.

If the next character is none of the above, the RAM flag is set and the counter zeroed in preparation for counting the characters to be underlined. Finally, the RET at line 390 functions to ignore the "at" symbol and go back to Scripsit to get the next character.

Lines 400–520: Before every character is printed, this subroutine checks the flag state. If it is not set, the character is printed. If it is, the RAM counter is incremented. Note thet 450 and 460 perform one of the modification's idiot-proof functions: If you forget to turn off the underlining before you get to the end of a text line, this code keeps the printout from totally bombing.

Lines 580–800: This is the actual underlining routine. The count from the RAM counter is placed in the B register, the backspace character (08H) is placed in the C register, and the printer is backspaced. Next, the underscore character (5FH) is placed in the C register and printed until the printer carriage returns to its former position. Finally, program control is returned to the main body of Scripsit to get the next character.

While this may seem like a long and complicated routine, you won't actually notice any slow-up in your printing. This is because any word processing printer is almost infinitely slower in accepting characters than the computer is in sending them. If you don't believe me, try eliminating the printer status check and loop-till-ready subroutine at lines 770–800 and see what happens.

Using Scripmod

Your newly modified Scripsit program will now give you total control over the following areas:

1) Printing lines (underscore characters)
2) Underlining text
3) Pausing the printer (to change type fonts, etc.)

Using these controls requires certain modifications to the original text and involves certain limitations.

Printing Underscore Characters

To print a line of underscore (5F hex) characters, simply type a series of "at" characters (shift 0). The only limitations are that the line must be at least two characters long and the last character must be followed by a space or a line terminator (see Scripsit instruction manual for the definition of a line terminator).

Underlining

Underlining text requires placing the underscore control character (shift 0) immediately before and immediately following the word or phrase to be underlined. To have your text print "Atlas Shrugged is a book by Ayn Rand," you would type: (shift 0) Atlas Shrugged (shift 0) is a book. . .

There are several important limitations of the underlining mode:

- The first character following the initial "at" symbol must be a printable character. It must not be a space, a line terminator or a control character.
- The text to be underlined must not be formatted as justified or flush right.
- If the phrase to be underlined extends to more than one line of printed text, each line must have the underlining terminated and the next line must restart the underlining anew. Be especially careful when reformatting or adding text to underlined areas; you may inadvertently change your line structure.
- If your printer has a text buffer that is limited to one line of type, the phrase length of underlined text may be limited. Remember: the actual line length is equal to the number of characters *plus* the number of backspace characters *plus* the number of underscore characters. What happens after the buffer overflows is dependent on your particular printer.

Pausing the Printer

When you want the printing operation to pause in the text, type an "at" character followed immediately by a greater than symbol (>). No message will be printed on the screen; the printer will simply halt. To resume printing, press Enter or the space bar.

A few caveats are in order: most printers that use a print buffer will not print to the middle of a line before pausing. They will hold the contents of the buffer until printing is resumed. Consequently, it will usually be necessary to plan your type font change to coincide with a new print line.

Because the program is now looking for the new control character combination to halt printing, phrases to be underlined must not begin with the greater than symbol.

Some printers (Anadex is the only one I know of) dump their print buffer after approximately 10 seconds of not receiving new characters. (Don't ask me why!) Obviously, this modification cannot be used with such printers.

A Few Last Words

By taking on the somewhat tedious task of trying to follow the program flow of this Scripsit modification, you will no doubt learn quite a bit about assembly/machine language programming. You will have an increased appreciation for how Scripsit actually works, and thus be able to make some of your own modifications.

Any questions or comments regarding the modifications are most welcome. For a fee I will also alter your Scripsit disk for you.■

Program Listing 2

```
                00090 ;          PROGRAM 2:
                00100 ;          MODULP/CIM  - UNDERLINING SCRIPSIT
                00110 ;          (MUST MOVE SCRIPSIT TO 8200H BASE ADDR FIRST)
                00120 ;          START=8200 END=AB40 TRA=AB26
                00130 ;
7B26            00140 FLAG       DEFL    RELO-3000H        ;RAM FLAG
7B28            00150 CNTR       DEFL    FLAG+2            ;RAM COUNTER
AA9E            00160            ORG     0AA9EH            ;MOD STARTS HERE
AA9E FE40       00170            CP      40H               ;CHAR = @?
AAA0 2809       00180            JR      Z,USCR            ;CO IF IT IS
AAA2 CDCD7A     00190            CALL    FLGST-3000H       ;CHECK FLAC STATUS
AAA5 32E837     00200 PRT        LD      (37E8H),A         ;PRINT CHAR
AAA8 C3745F     00210            JP      5F74H             ;RETURN TO SCRIPSIT
AAAB 08         00220 USCR       EX      AF,AF'
AAAC 3A267B     00230            LD      A,(FLAG)          ;CHECK FLAG STATE
AAAF B7         00240            OR      A
AAB0 2866       00250            JR      Z,STPUL           ;GO IF SET
AAB2 7E         00260            LD      A,(HL)            ;GET NEXT CHAR
AAB3 FE3E       00270            CP      3EH               ;CHAR = >?
AAB5 2866       00280            JR      Z,PSTOP           ;PRINT IF IT IS
AAB7 FE40       00290            CP      40H               ;CHAR = @?
AAB9 2828       00300            JR      Z,PRUL            ;PRINT IF IT IS
AABB FE21       00310            CP      21H               ;CHECK FOR SP, CR, LF
AABD 3824       00320            JR      C,PRUL            ;PRINT IF IT IS
```

Program continues

✓See List of Advertisers on page 306

```
AABF FE80          CP      80H            ;CHAR = CODED SP?
AAC1 3020          JR      NC,PRUL        ;PRINT IF IT IS
AAC3 3E00          LD      A,00H
AAC5 32267B        LD      (FLAG),A       ;SET FLAG
AAC8 32287B        LD      (CNTR),A       ;ZERO THE COUNTER
AACB 08      GTCHR EX      AF,AF'
AACC C9            RET                    ;(TO SCRIPSIT)
AACD 08      FLGST EX      AF,AF'
AACE 3A267B        LD      A,(FLAG)       ;GET FLAG STATUS
AAD1 B7            OR      A              ;CHECK SET
AAD2 200D          JR      NZ,NTST        ;GO IF NOT SET
AAD4 08            EX      AF,AF'
AAD5 FE0E          CP      0EH            ;CHAR = CR, LF, ETC?
AAD7 3814          JR      C,ULRTN        ;START UNDERLINING
AAD9 08            EX      AF,AF'
AADA 3A287B        LD      A,(CNTR)
AADD 3C            INC     A              ;INCREMENT COUNTER
AADE 32287B        LD      (CNTR),A
AAE1 08      NTST  EX      AF,AF'
AAE2 C9            RET
AAE3 3EFF    PRUL  LD      A,0FFH         ;RESET FLAG
AAE5 32267B        LD      (FLAG),A
AAE8 08            EX      AF,AF'
AAE9 3E5F          LD      A,5FH          ;CHANGE TO UNDERSCORE
AAEB 18B8          JR      PRT            ;PRINT IT
AAED 08      ULRTN EX      AF,AF'
AAEE C5            PUSH    BC
AAEF 0E08          LD      C,08H          ;BACKSPACE CHAR
AAF1 CD017B        CALL    CNTDN-3000H    ;BACK UP TO WORD START
AAF4 0E5F          LD      C,5FH          ;UNDERSCORE CHAR
AAF6 CD017B        CALL    CNTDN-3000H    ;UNDERLINE WORD
AAF9 3EFF          LD      A,0FFH         ;RESET FLAG
AAFB 32267B        LD      (FLAG),A
AAFE C1            POP     BC
AAFF 08            EX      AF,AF'
AB00 C9            RET
AB01 3A287B  CNTDN LD      A,(CNTR)       ;GET COUNT
AB04 47            LD      B,A
AB05 79      PRCHR LD      A,C            ;GET CHAR
AB06 32E837        LD      (37E8H),A      ;PRINT IT
AB09 CD107B        CALL    PRSTAT-3000H   ;CHECK PRINTER STATUS
AB0C 05            DEC     B              ;REDUCE COUNT
AB0D 20F6          JR      NZ,PRCHR       ;LOOP IF NOT ZERO
AB0F C9            RET
AB10 3AE837  PRSTAT LD     A,(37E8H)      ;GET STATUS CHAR
AB13 CB7F          BIT     7,A            ;TEST IT
AB15 20F9          JR      NZ,PRSTAT      ;LOOP IF NOT READY
AB17 C9            RET
AB18 08      STPUL EX      AF,AF'
AB19 CDED7A        CALL    ULRTN-3000H    ;GO TO UNDERLINE
AB1C C9            RET
AB1D 23      PSTOP INC     HL             ;ADVANCE CHARACTERS
AB1E 3A4038  PLOOP LD      A,(3840H)      ;CR ON KEYBOARD?
AB21 B7            OR      A
AB22 28FA          JR      Z,PLOOP        ;LOOP IF NOT
AB24 08            EX      AF,AF'
AB25 C9            RET
             ;     END OF UNDERSCORE ROUTINE
             ;     RELOCATOR CODE FOLLOWS
AB26 21E303  RELO  LD      HL,03E3H
AB29 221640        LD      (4016H),HL
AB2C F3            DI
AB2D 210082        LD      HL,8200H       ;START ADDR
AB30 110052        LD      DE,5200H       ;DEST ADDR
AB33 012729        LD      BC,RELO-81FFH  ;BYTE COUNT
AB36 EDB0          LDIR
AB38 3EFF          LD      A,0FFH         ;RESET FLAG
AB3A 32267B        LD      (FLAG),A
AB3D C30052        JP      5200H
AB40 00      FINIS NOP     ;JUST TO QUICKLY FIND PROGRAM END
0000               END
00000 TOTAL ERRORS
```

FOR TRS-80*

SUPERSCRIPT

By Richard Wilkes

Now you can easily print **BOLDFACE**, underlined text, slashed Ø's, special characters, superscripts & subscripts (A^{c} X $\log_{e}[A_{1}-A_{n}]$) and more!

With this enhancement to Radio Shack's Model I Scripsit,* you will turn a good word processing system into a great one. SuperScript adds many of the features that Fort Worth left out, including:

Custom print drivers for:
Diablo (serial or parallel)
NEC Spinwriter (serial or parallel)
RS Line Printer IV*/Centronics 737
RS Daisy II*
Standard serial and parallel printers
User-defined custom drivers (serial or parallel)

You can call up the disk directory or kill files without leaving Scripsit—and without losing your text! Pause the printout to insert text from your keyboard or change type wheels, then resume printing where you left off.

Using any printer with backspace capability, you can underline text and produce computer-type slashed zeros. And on Diablo, Daisy II and NEC printers, you can superscript, subscript, underline, print boldface and select 10- or 12-pitch. With the Line Printer IV/737, you can underline, print elongated boldface, select 10 or 16.5 pitch, and use proportional spacing (unjustified).

The keyboard driver is now modified to correct for repeating key hangups. You may specify spacing requirements to eliminate awkward spacing of critical text. And, with SuperScript, you can now enter special characters (brackets, braces, etc.) that are not found on the TRS-80* keyboard.

All these capabilities, and more, are available when you add SuperScript to your Model I Scripsit program. Available for just $29.95 on disk, including easy instructions for patching to Scripsit and an enhanced lowercase driver.

SYSTEM SAVERS

By Tom Stibolt

If you ever type "SYSTEM" on your TRS-80,* this two-program package will make life easier for you.

One of the programs, FLEXL, lets you make backup copies of any system format tape. Using your own recorder usually means easier loading than with machine-duplicated original tapes, and you will be able to store your original safely away. Copies made using FLEXL display the filename of each program as it loads, making file searches easier.

Disk drive owners can use TDISK to save any system format tape onto disk. "Editor/Assembler", "Air Raid" and other programs cannot normally be loaded to disk, but with TDISK, they can. It will even load non-contiguous tapes. Why put up with slow tape loading? TDISK files will load from disk in seconds.

Get this two-program package now for only $14.95. Just one of Acorn's fine utility programs.

* trademark of Tandy Corp.

Acorn
Software Products, Inc.

634 North Carolina Avenue, S.E., Washington, D.C. 20003 ✓34

These and other popular Acorn programs are available now at fine computer stores. Ask for them.

DEALER INQUIRIES INVITED

More tricks of the trade for savvy programmers.

Program Chaining and Local Variable Definitions in BASIC

Hal Brown
643 W. Valley Forge Road
King of Prussia, PA 19406

Chaining, in computerese, refers to loading and running a series of programs, with each picking up data where it is left by the previous program. After doing its thing, one program leaves updated values for the next program. The programs themselves normally overlay in memory—if they didn't there would be little point in chaining; the functions of the different programs could all be provided in one large program.

Pointers

Unfortunately, chaining normally is impossible in TRS-80 Level II or Disk BASIC. Execution of a LOAD (or CLOAD) or RUN command initializes some pointers in scratch memory. In particular, three pointers are involved. The pointer to the start of simple variable storage area in memory is at 40F9H. 40FBH points to the beginning of array variable storage and 40FDH points to the lowest byte in free memory.

At initialization the three pointers are loaded with the address of the first byte past the end of the BASIC program in memory. With the top and bottom of variable storage indicated as the same address by the pointers, the BASIC interpreter assumes that no variables are stored. Requests for variable values immediately after initialization return zeros (nulls for strings), so values developed during execution of one program are lost when another is loaded and run.

It is possible to transport variables from one program to another by saving them on tape or disk at the end of the first execution and reading them back at the beginning of the next, but that can be time-consuming if a lot of data is involved.

If all you want to do is rerun the same program or a different segment of the same program without zeroing the variables, this is easy. The secret is to use GOTO instead of Run. For example, assuming line 10 is the first in the program, GOTO 10 has the same effect as Run except that variables defined in a previous run are preserved.

Variable Preservation

It is not as easy to preserve variables between loads and runs of different programs, but it can be done with the help of a few additional instructions. The method I'm about to describe tricks BASIC into storing the variables in a different area from that indicated by the pointers during initialization.

Conceptually the technique is simple. Near the beginning of a program run, new addresses are POKEd into the three pointers described above. BASIC will store variables defined after that at the new place in memory. At the end of the program, the updated pointers are preserved in a safe place in memory and the original pointers restored.

Each program in the chaining set can retrieve the addresses left by the previous program and POKE them into the 40FnH pointer locations, thereby recovering the variables for further use and processing.

There are pitfalls to avoid in implementing this method. Remember that as soon as the pointer addresses are changed, BASIC will be unable to find variables defined before the change. To illustrate how this can be a problem, consider the way the block move of the three addresses (six bytes) from memory to the pointers might be accomplished. Since the pointers are adjacent in scratch memory, the first inclination is to use a FOR-NEXT loop. It won't work! Somewhere in the middle of the instruction, BASIC will be unable to find the operand (the "I" in a FOR I = n TO m instruction) and execution will abort.

The pointer moves could be accomplished with a small machine language routine called by a USR instruction. I tried that first. The routine was packed into a string so no memory had to be reserved and it worked well. However, I finally opted for the approach used in Program Listing 1. It may not be as sophisticated as the use of string-packed machine language routines, but it works.

Variable Storage

Implementing transportable variables as discussed here requires preselection of an area in memory where the variables will be stored. Some care is required in the selection. If it is too low in memory, the longest program in the chaining set (or perhaps a longer one yet to be written) could overwrite it. If too high in memory, the string literals or stack may do the same.

It may seem that a location in reserved memory, above memory size, would be a good choice. Just set a relatively low memory size when entering BASIC, then load the variable storage pointers with a higher address. Unfortunately this won't work. Whenever a new variable is to be stored, BASIC checks the free-space pointer. If it points to an address in reserved memory, an out-of-memory message results.

A location just a little above the longest program to be chained would be ideal, but usually there will be no problem allowing generous space above the longest program for future expansion.

For the demonstration program, I selected addresses in middle memory—well above the end of the program—so those who want to experiment with the concept can do so without changing the locations. Longer segments can be written to replace the five line routines that currently represent the processing portions of the program.

String Storage

In addition to relocating the variable storage area pointers, attention must be given to one

other detail if string variables are to be transported from one program to another. It is related to the way strings are stored. In the case of numeric variables the values along with their attributes (type and name) are saved in the variable storage area. This is not true for strings. Only the attributes plus a pointer to the string literal (the actual string text) are saved in the variable storage area. If completely delimited by quotes within the program, string literals are left there; but all others are stored in high memory starting just below reserved memory and expanding downward.

A pointer at 40D6H directs data to the next string literal storage space, i.e., to the address just below the most recently stored string. (40D6H has been incorrectly identified as the storage space for memory size in some memory maps. It points to the address just below memory size only until the first string literal is stored there.) To prevent writing over existing strings in high memory with strings saved after a program change, it is necessary to save and rePOKE this pointer value since it will also be reinitialized during the load and run.

While discussing transportable string variables, note one limitation. If the string is completely defined in the program in a statement such as S$ = "THIS IS THE STRING LITERAL", it will not be stored in high memory. Instead the pointer in the variable storage area points to the string in the program area.

Strings so defined will be destroyed when another program is loaded. Thus the restriction: strings to be transported between programs must not be wholly defined in a simple assignment statement in the program. To circumvent this limitation, the string above could be defined as S$ = "THIS IS " + " THE STRING LITERAL". The two parts will be concatenated by BASIC and the result stored in upper memory where it can be protected.

I noted earlier that the original variable storage pointer is saved and restored at the end of a program run. The need for this may not be obvious. Only the pointer at 40F9H need be restored, but failure to restore it can produce strange effects. Once while I was developing the demonstration program, this pointer was not replaced. It remained pointing to an address in high memory. Since the program seemed to be working, I saved it on disk. Subsequently, after noting peculiar repeatable conditions in middle memory following a program load, I determined that the save had recorded everything from the first line of the program to the high address pointed to by 40F9H. The Save (and CSave) routine uses the 40F9H pointer address as the end-of-file pointer.

One other point: even if there is no other reason to set memory size, TRSDOS 2.2/2.3 users should protect some space. During Disk BASIC loading, the system uses the top 64 bytes of RAM. Setting memory size at least 64 bytes below top-of-memory will prevent loss of some strings if it is necessary to reload Disk BASIC following DOS reboot.

Initialization

Special initialization is required for the transportable variables system to work. Since addresses will be recovered from designated memory locations and POKEd into pointers, addresses must be available for recovery. They will be available following the first execution of a program in the chaining set, but not before that. Consequently an initialization routine must be executed at the beginning of each chaining session. The routine loads initial pointer addresses into the designated memory locations for recovery by the first program execution.

To review briefly, the following steps must be taken during execution of a program to recover previously defined variables, process them and save the updated values for use by the next program:

1) Store the pointer value at 40F9H in a safe place in memory.
2) Retrieve the string pointer address from memory and POKE into 40D6H.
3) Retrieve three other pointer addresses and POKE into 40F9H, 40FBH and 40FDH.
4) Now use and process the variables as desired.
5) Before terminating the run, store the updated pointers in memory.
6) Store the updated string pointer.
7) Retrieve and replace the original value in 40F9H.

An initialization routine is appended to the end of the demonstration program starting at line 1000. In a practical application, this routine could be appended to every program in the set so any one of them could be loaded and run first, or it might be incorporated in a special program that is executed at the beginning of each chaining session.

Once the ability to arbitrarily select variable storage locations in memory has been developed, there is nothing in principle to prevent defining a second, third, or more, each for use in a different segment of the program. Within each segment BASIC will be unaware of variables defined in another segment. Consequently, variables in different segments of the program can be given the same names even though they are completely independent.

You may recognize this as the much heralded capability to define local (as opposed to global) variables. This is one of the important features of structured high level languages. (Move over Pascal!)

The problem with this method of providing local variables is that only local variables are available—none are global. As

```
10 '  TRANSPORTABLE and LOCAL-GLOBAL VARIABLES
20 '           DEMONSTRATION PROGRAM
30 '
40 '                by Hal Brown
50 '           643 W. Valley Forge Rd.
60 '          King of Prussia, PA 19406
70 '
80 CLEAR200:DEFINTQ
90 Q=32250:POKEQ-1,PEEK(16634):POKE Q-2,PEEK(16633)
100 POKE16599,PEEK(Q-3):POKE16598,PEEK(Q-4)
110 GOSUB 300
120 B=0:QV=VARPTR(B):Q=32500:GOSUB500
130 G$="":QV=VARPTR(G$):GOSUB500
140 PRINTA;TAB(6)B;TAB(15)S$;TAB(33)G$
150 A=A+1:B=B+2:S$=S$+"S":G$=G$+"G"
160 PRINTA;TAB(6)B;TAB(15)S$;TAB(33)G$
170 Q=32250:COSUB400:Q=32500:GOSUB300
180 B=0:QV=VARPTR(B):Q=32250:GOSUB500
190 G$="":QV=VARPTR(G$):COSUB500
200 PRINTA;TAB(6)B;TAB(15)S$;TAB(33)G$
210 A=A-2:B=B+3:S$=S$+"X":G$=G$+"V"
220 PRINTA;TAB(6)B;TAB(15)S$;TAB(33)G$
230 Q=32500:GOSUB400
240 Q=32250:POKEQ-3,PEEK(16599):POKEQ-4,PEEK(16598)
250 POKE16634,PEEK(32249):POKE16633,PEEK(32248)
260 END
300 POKE16621,((QAND-256)-2*(QAND-32768))/256
310 POKE16620,QAND255
320 POKE16633,PEEK(Q)
330 POKE16634,PEEK(PEEK(16620)+256*PEEK(16621)-512*(PEE
     K(16621)AND128)+1)
340 POKE16635,PEEK(PEEK(16620)+256*PEEK(16621)-512*(PEE
     K(16621)AND128)+2)
350 POKE16636,PEEK(PEEK(16620)+256*PEEK(16621)-512*(PEE
     K(16621)AND128)+3)
360 POKE16637,PEEK(PEEK(16620)+256*PEEK(16621)-512*(PEE
     K(16621)AND128)+4)
370 POKE16638,PEEK(PEEK(16620)+256*PEEK(16621)-512*(PEE
     K(16621)AND128)+5)
380 RETURN
400 FORQI=0TO5:POKEQ+QI,PEEK(16633+QI):NEXT:RETURN
500 QE=PEEK(Q+2)+256*PEEK(Q+3)-512*(PEEK(Q+3)AND128):QA
     =Q+6
510 IF(QA-2*(QAAND-32768))>=(QE-2*(QEAND-32768))THENRET
     URN
520 FORQI=0TO2
530 IFPEEK(QA+QI)<>PEEK(QV-3+QI)THENQA=QA+PEEK(QA)+3:GO
     TO510ELSENEXT
540 FORQI=0TOPEEK(QV-3)-1:POKEQV+QI,PEEK(QA+3+QI):NEXT:
     RETURN
1000 Q=32250:GOSUB1100
1010 Q=32500:GOSUB1100:END
1100 QH=(Q-2*(QAND-32768))/256:QL=(QAND255)+6
1110 IFQL>255THENQL=QL-256:QH=QH+1
1120 FORQQ=QTOQ+4STEP2
1130 POKEQQ,QL:POKEQQ+1,QH:NEXT
1140 POKE32246,PEEK(16598):POKE32247,PEEK(16599)
1150 RETURN
```

Program Listing

soon as a new segment of the program is entered by changing the variable storage pointers, all variables defined in the previous segment are effectively non-existent.

Global Variables

Ability to define local variables is of little value without an accompanying ability to define global variables, so a means of providing the latter is needed. The program demonstrates the technique devised.

Any non-array variable can be made global using the routines in the current program. Array variables are stored in different format, but could be made global using the same general technique.

The means used in one segment of the program to obtain a variable from another is illustrated in line 180 of the demonstration program. B is the variable to be made global. First it is defined with a default value. If the variable exists in the program segment to be interrogated, the default value becomes a dummy and could be anything, but it will be the value assigned if the specified variable is not found in the other segment. In the default assignment a variable must be the same type—integer, single precision, double precision or string—and have the same name as the variable in the other segment; otherwise it will not be found.

Following the default value assignment a VARPTR statement assigns the address of the variable to QV. Q is then assigned the address of the other segment's variable storage area and a GOSUB 500 completes the recovery.

Subroutine 500 compares the first three bytes of each variable storage assignment in the other segment's storage area with the first three bytes for the variable in the current segment storage space. These bytes specify the type and name. If a match is found, the contents of the remaining bytes in the variable's assignment space are loaded from the other segment space into the current space. The variable now exists in the new segment with the same value it last had in the previous segment. If no match is found before reaching the address of the start of array storage space, execution returns to the subroutine calling point and the default value remains assigned to the variable.

This procedure can be repeated for as many variables as desired. For a subsequent run of the program, the updated value of the global variable is used in the first segment of the program, and there must also be a variable transfer sequence included in that segment. This was done in the demonstration program.

If this explanation of the global-variable definition has confused you, welcome to the club. I'm confused and I wrote it! A study of the statements in the program will hopefully clarify it.

The current program is a compromise designed for non-disk owners *and* for disk system owners. To demonstrate both the ability to transport variables from program to program and the local-global variable capability, I had to identify two variable storage areas. To keep the demonstration within the limits of a 16K system, available memory for the purpose is limited, especially for disk users. DOS occupies most of memory up to 7000H, and by the time space is allowed for the BASIC program, there isn't room for large variable storage areas. However, large areas aren't needed for the demonstration.

The addresses selected are nominally at 32250 (7DFAH) and 32500 (7EF4H). If more extensive experimentation is to be done or a practical application made of the concept, Level II users with 16K may want to use lower addresses and Disk BASIC users with added memory can specify higher addresses.

Address Changing

Address changing is relatively easy. Note all the places in the program where Q is assigned a value. It will be either 32250 or 32500. To change one of these, change each occurrence of the number to the new address. Remember to use the negative form for addresses above 32767

(7FFFH). In addition, if 32250 is the address being changed, the two related addresses in line 250 and two in line 1140 must be changed to their equivalents relative to the new address. Variable expressions Q-1, Q-2, Q-3 and Q-4 could not be used at these points because the variable area pointer is being changed and BASIC would be unable to find Q.

When selecting an address for relocating variable storage, I suggest that the address not be near the address sign-change point, −32768 or 8000H. At several places in the program, addresses such as Q-2 or Q+6 are specified. If Q equals −32767, then Q-2 is outside the integer range and an error message will result. Similarly, if Q equals 32767, Q+6 is outside the integer range.

I designed the program for easy conversion into a longer demonstration program or into the separate programs of a useful chaining set. The two processing segments of the program each consist of five lines including the global variable definition lines. The first includes lines 120–160, the second lines 180–220. By replacing either block of lines with a routine of your own, the program can be customized.

If the local-global variable feature is unnecessary, the second set of processing lines and associated subroutine calls can be deleted. The several programs that comprise a chaining set could all be produced by inserting their routines into the middle of this demonstration program.

Program Use

To use the program, after typing it in and saving on tape or disk, first type RUN 1000 (enter). This initializes the pointer addresses at the two selected variable storage areas. Now type RUN (enter). Four lines of values are displayed. The first two are the before and after processing values of the variables in the first segment, the second two are the same in the second segment. Two numeric and two string variables are displayed, although the strings won't appear on the first printouts from the two segments because their initial values are null.

The first number displayed on each line is the value of local variable A. The second is the value of global variable B. The first string is local S$, the second global G$.

With successive runs of the program you will note that A is simply incremented with each execution of the program in the first segment, but the value representing the same variable name in the second segment is decremented by two with each program run. These two local variables maintain individuality despite their common name.

Global B is the same variable in both segments. It is incremented by two in the first segment and by three in the second. Note that in this case each segment picks up the value where it was left by the last segment executed.

The distinction between local and global variable treatment is especially evident in the string displays. Local string S$ has an S appended with each run of the first segment, while independent S$ in the second segment appends an X with each run. In the case of global G$, the first segment appends a G, the second a V, resulting after several runs in a string of alternate characters instead of the repeated same characters of the local strings.

The variables will survive a NEW command and reload of the program (do not reinitialize with a RUN 1000 following reload unless you desire to zero the variables). They will even survive DOS reboot and reload of Disk BASIC provided you heed my warning about the top 64 bytes of memory (TRSDOS 2.2/2.3 only). To simulate the chaining of different programs, processing algorithms can be changed and different versions independently saved, or just change them between runs. Program editing will not affect the variables safely stored in middle memory.

Points of Interest

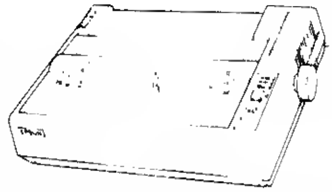
Several aspects of the program may be of interest. The let-

Continues to page 263

BUSINESS COMPUTERS

*TM TANDY CORP.

THE MX-80 NOT ONLY DOES EVERYTHING, IT DOES EVERYTHING WELL.



This is the new Epson MX-80 dot matrix printer. It does just about everything you could ask a printer to do. Quickly. Quietly. Reliably. In fact, for OEM installations, the MX-80 may be the single best, all-round printer you can buy. But that's not the best reason to buy it.
The MX-80 prints bidirectionally at 80 CPS in a user-defined choice of 40, 80, 66 or 132 columns. And if that's not fast enough, its logical seeking function minimizes print head travel time. The MX-80 prints 96 ASCII, 64 graphic and eight international characters with a tack-sharp 9x9 matrix. For a long time. Epson printers are known for reliability and the MX-80 is no exception. But that's not the best reason to buy it either.
The print head has a life expectancy of up to 100×10^6 characters, and when it wears out, just throw it away. A new one costs less than $30 and the only tool you need to change it is attached to the end of your arm. The MX-80 is compact weighs only 12 lbs., and the whole unit, including the two stepper motors controlling carriage and paper feeding functions, is precisely controlled by an internal microprocessor. But even that isn't why you should specify the MX-80.
The best reason is this: because Epson makes more printers than anyone else in the world, we can afford to sell each one for a little less.

...Call For Unbelievably 'Low Price

525

OKIDATE MICROLINE 80 PRINTER - $479

The Microline is built on a rugged cast aluminum base to withstand the rigors of continuous business use. It is driven by two motors and will run all day with no duty cycle limitations. Microline printers use a dense 9 x 7 dot pattern to produce crisp, clean copies, first copy to last. The seven pins in the head are "fired" using energy stored in tension members. This technology permits the use of short, low mass pins made with an extremely hard alloy. The head produces less heat, thereby extending its life. 524

AW...WHAT THE HECK
RAM Memory Chips for the TRS-80

It is the policy of American Business Computers to offer merchandise at the lowest price possible. Serveral months back we began selling RAM Memory Chips for the TRS-80 for $45.00 per set. Someone else sold chips for $44.00. We sold them for $38.00. They sold them for $37.95. So we say "AW ..... WHAT THE HECK!" Let's see the other guys beat this price.

$22

PER 16K SET

These chips are brand new "4116's". These 200 nanosecond chips are fully compatible with all TRS-80 produces. Instructions for insertion are included, however the dip shunts required for converting a 4K Model I to a 16K Model I are not included at this low price.

526

★ Check It Out! ★

EPSON MX-80FT

That's right — **MX-80FT.**
The **FT** on the end means Friction and Tractor. The Fantastic **MX-80** Printer is now available in a version which will accept letterhead-type paper **And** tractor-type paper. Call or write for our (as always) unbelievable low price.

527

This is the Epson MX-70. The lowest priced dot matrix printer you can buy. Now, that in itself should make it very attractive to a lot of people. But you ain't heard the half of it.

To begin with, the MX-70 has a lot more in common with our now-famous MX-80 than just the name. Like unequalled Epson reliability. And technological breakthroughs like the world's first disposable print head. But frankly, the MX-80 packs a lot more power than some people need. So Epson built the MX-70 to be a no-frills printer. At a no-frills price.

But the MX-70 is still a great little printer. They give you 80 CPS unidirectional printing. Top-of-form recognition. Programmable line feed and form lengths. Plain paper printing. An easy-to-read 5x7 matrix. Self test. And an adjustable tractor feed.

That's what you'd expect from a basic little printer. But here's something you wouldn't expect: the finest graphics package on the market today. Free.

They call it GRAFTRAX II. And it means 480 dots across the page, resolution to 60 dots per inch, and a graphic image free of the jitter and overlap that plagues other printers. You get cleaner grays and finer point resolution.

So now you've got a choice. You want more power and extra functions, you buy the MX-80. You want a basic little printer that prints, and keeps on printing, you buy the MX-70. They're both at American Business Computers.

Call for unbelievably low price.

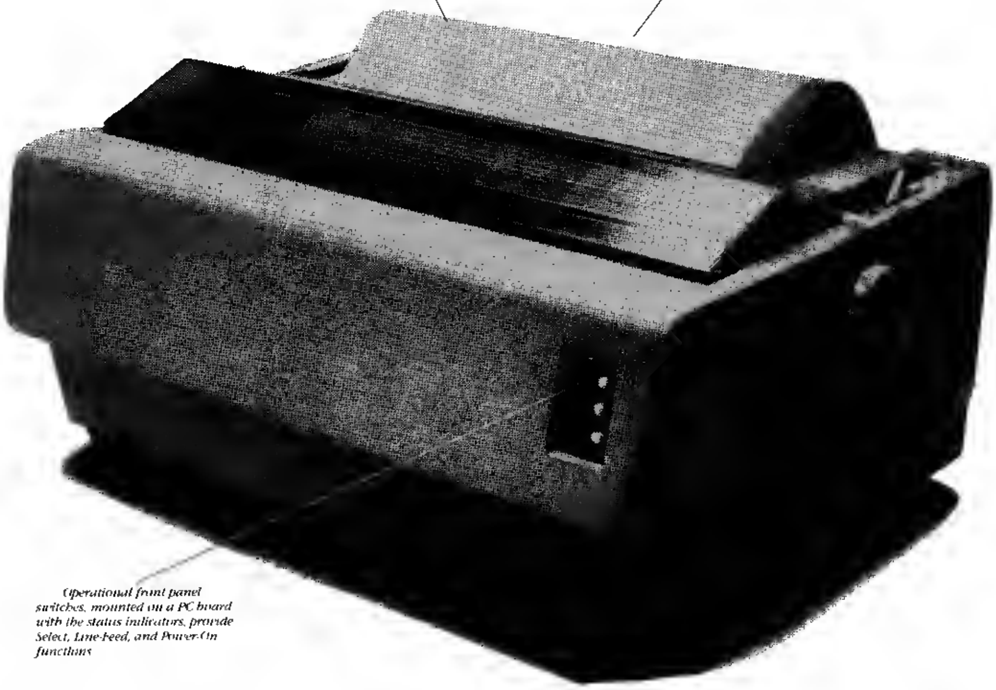
Epson.

This printer costs less than $450. just how much less we can't tell you - But if you will give us a call we think you will be pleasantly suprised.

528

NEED A LETTER QUALITY PRINTER BUT DON'T WANT TO PAY THE PRICE?

Then American Business Computers Has Just The Printer For You!



The Starwriter Daisy Wheel Printer is designed to fill a distinct gap in the peripheral marketplace. A speed of 25 cps and its low price make it the ideal choice for today's systems designer interested in achieving superior price/performance ratios.

Incorporating the latest LSI technology, the Starwriter is built by one of the world's most respected computer peripheral manufacturers. The printer is furnished complete and ready-to-use, requiring no changes in hardware or software. Starwriter offers an industry standard parallel interface, as well as RS232-C interface. Total plug compatibility and a wide variety of interface matching capabilities help lower the system integration costs for OEMs and end-users.

Using a 96-character wheel, the Starwriter produces letter-quality printing on 3 sharp copies with up to 136 columns. Starwriter offers the highest degree of vertical and horizontal positioning, resulting in the most precise character placement in the industry. The easy-to-change character wheel also makes our printer a perfect choice for international applications. Compatible with sheet feeders, starwriter accommodates paper widths to 381 mm (15") and uses industry standard ribbon cartridges.

Along with a self-test capability and a programmable VFU (Vertical Format Unit), Starwriter furnishes the operator with all desirable status functions, commands and program selectable switches. Panel lamps indicate the current Paper, Select and Power status.

The easy plug-in compatibility of Starwriter and its outstanding print performance makes the Starwriter perfect for TRS-80*, Apple*, and S-100 users.

Starwriter is backed up by a one-year warranty (3 months on parts and labor, 9 succeeding months on parts) and is further supported by a strong, fully trained and technically proficient service organization. The Starwriter is available for immediate shipment in any quantity desired.

For more information, call American Business Computer, 118 So. Mill Street, Pryor, Oklahoma 74361; Telephone 918-825-4844.

*TM of Tandy Corp
*TM of Apple Computer Corp.

AMERICAN BUSINESS COMPUTERS ✓529

118 So. Mill Street
Pryor, Oklahoma 74361
Telephone 918-825-4844

Dealer Inquiries Invited

ter Q was used as the first letter of all housekeeping variable names on the assumption that it is unlikely to be a variable name in a user routine. Since all the housekeeping variables are integer, all Q name variables are defined integer with the DEFINT statement.

Note that when new addresses are loaded into the 40FnH pointers via subroutine 300, a complex system of POKEs is required in lieu of a FOR-NEXT loop, to circumvent the problem mentioned earlier. However, FOR-NEXT works just fine when the updated pointer addresses are saved in memory via subroutine 400.

The CLEAR 200 at the beginning of the program requires a little discussion. If local variable strings are defined in two or more independent segments of a program, a special problem can occur after a number of runs. It results from the way strings are stored in memory. As noted earlier, in the case of string variables only the characteristics and a pointer are stored in the variable storage area. Most string literals are stored in high memory.

When BASIC stores string literals each newly defined string is placed in free space below the last stored string, even though the new string may be only a redefinition of a string variable and therefore a replacement for a string literal already stored. This continues until there is insufficient space left in the region reserved for strings to store the next defined string.

At this point a garbage detail in BASIC determines which literals still represent active variables in the program and which do not. The active literals are repacked into space starting at the top of the string area and the rest of the space is made available for new string storage. Generally this will produce sufficient space to store the new string literal that initiated the cleanup. If not, an out-of-string-space error results.

The normal way to correct this error is to add a CLEAR nn statement or increase the size of the one already present. When no CLEAR nn statement is provided, the default value of 50 is assumed. If independent segments of a program are defined as described in this article, a problem potentially more serious than an error message can result. When the garbage cleanup occurs in one segment, the local variable strings defined in another segment are unknown to the garbage detail. As a result, they may be overwritten in the process of repacking the strings that are known.

There is no completely safe way to avoid this problem. However, specifying a relatively large CLEAR nn space can eliminate it for most applications. The CLEAR 200 statement in the program will delay its occurrence until after execution of many more runs than required to demonstrate the features of interest. However, try some runs with the CLEAR 200 statement deleted to see that the problem is evident after just a few runs.

This potential overwrite problem affects only strings, and only when two or more independent program segments are defined. It is not a limitation on simple variable transportation for chaining, which is considered the most useful of the several features demonstrated.

Logical AND expressions are used in program lines containing conversion of addresses to or from the most and least-significant byte components. With the variable storage area addresses currently selected for the program, these logic expressions are not needed. However, when users with more than 16K of memory use addresses above 7FFFH, there can be a problem. Such addresses must be expressed in negative form in integer expressions or when used in PEEK and POKE statements. On the other hand, the component byte values of all addresses are retrieved from memory as positive integers and must be POKEd as positive numbers.

The logic expressions automatically take care of these sign changes when needed. Initially, you must specify high addresses in negative form (or in hexadecimal with Disk BASIC), but the AND expression will take care of required sign changes after that. ■

✓See List of Advertisers on page 306

Free valuable disk space by eliminating DOS.

KILDOS Is Here

Robert Soltysik
2613 Evans Ct.
Plano, TX 75075

There comes a time when a single disk drive is not enough. When you do finally part with the cash for a new drive, you have a large task ahead of you organizing all your existing disks.

My nature being what it is, cheap, I wanted to get all I could on each disk. When I got my second drive, I wanted to remove the DOS from all the disks but save the programs. I could then utilize the space previously occupied by the DOS for more programs. This article is about a program I wrote to kill the DOS.

BOOT/SYS	1 GRAN
DIR/SYS	2 GRANS
SYS0/SYS	3 GRANS
SYS1/SYS	1 GRAN
SYS2/SYS	1 GRAN
SYS3/SYS	1 GRAN
SYS4/SYS	1 GRAN
SYS5/SYS	1 GRAN
SYS6/SYS	3 GRANS
FORMAT/CMD	3 GRANS
BACKUP/CMD	3 GRANS
BASIC/CMD	4 GRANS
TOTAL	24 GRANS

Table 1. List of Extra Granules

Raquiras Apparat

The program, KILDOS (Listing 1), requires NEWDOS by Apparat. It will work with NEWDOS, NEWDOS +, or NEWDOS 80.

Three versions of the program, with slight differences, are necessary for killing the DOS on a disk with NEWDOS, NEWDOS 80 or TRSDOS. The changes are covered in Listings 2 and 3.

When you put a minimal DOS on a disk (by killing its utilities like DISKDUMP/BAS and TAPEDISK/CMD), you still leave a little over 20 granules filled with system "brains." When you have one disk drive, every disk you use must have brains on it to give the system instructions. When you have more than one drive, you may leave the brains in drive 0 to provide housekeeping information for all your disks that run on a separate drive.

Table 1 is a listing of the allocated space for a minimal DOS on TRSDOS. The allocation varies with operating systems, but you can see how much disk space can be gained.

A data disk needs only BOOT/SYS and DIR/SYS in any drive but drive 0. Obviously, more programs can be packed onto each disk if it contains no DOS.

How can we delete the DOS, but leave the programs? The simplest way is to KILL each system file one at a time. Since I had 40 full disks when my second drive arrived, the task would have taken a whole day. There had to be an automatic way to do it, so I wrote the program KILDOS.

Using tha Program

Insert a NEWDOS disk into drive 0 and the disk with KILDOS on it into another drive. KILDOS could also be on the NEWDOS disk in drive 0. (It is a good idea to always write protect the DOS disk in drive 0 for safety.)

Now boot the system, load BASIC, and then run the KILDOS program. It will respond with a question asking which drive you want to use for the killing process. After you answer, it will instruct you to load the disk to be deDOSed into the drive you specified.

When you hit ENTER you have begun the process; wait a moment and you will begin to see the results. The program lists the files it has killed, and, at the end, it tells you that you have no DOS and how many extra granules you have. The whole process takes about two minutes.

Once you have killed the DOS, you can proceed to fill the spaces with more files.

If you run into a glitch, the program will abort. If that hap-

```
5 REM   **** THIS PROGRAM BY R. SOLTYSIK  PLANO, TX.
10 REM  **** THIS MUST BE USED WITH 2 DRIVES
20 REM  **** FOR USE WITH INVISIBLE UTILITIES INTACT
30 REM  **** MAKES DATA DISK OUT OF SYSTEM (DOS) DISK
40 REM  **** KILLS ALL DOS EXCEPT BOOT AND DIRECTORY
60 REM  **** THIS PGM FOR TRSDOS 2.1, 2.2, 2.3
70 REM  **** YOU MUST USE NEWDOS IN DRIVE 0
80 CLEAR 5000: CLS: DIMA$(50): DEFINTA-Z:
90 I=0
100 INPUT "DOS KILLING PROGRAM. WHICH DRIVE WILL BE USE
    D"; D$
110 PRINT "INSERT DISK INTO DRIVE "D$ "   PUSH ENTER"
120 IFINKEY$<>CHR$(13)THEN120
130 CLS
140 FOR I = 1 TO 10: READ A$(I): NEXT I
150 DATA "SYS0/SYS","SYS1/SYS","SYS2/SYS","SYS3/SYS","S
    YS4/SYS","SYS5/SYS","SYS6/SYS","FORMAT/CMD","BACKU
    P/CMD","BASIC/CMD"
160 FOR J = 1 TO 10:C$="KILL "+A$(J)+":"+D$
170 CMD"C$"
180 PRINT A$(J)+":"+D$; "          KILLED"
190 POKE&H37E1,0
200 NEXT J
210 PRINT "   DONE    DISK CONTAINS NO DOS"
220 PRINT: PRINT "YOU NOW HAVE 21 MORE GRANS ON THIS DI
    SK"
```

Program Listing 1. KILDOS

pens, you can modify the DATA statements and the FOR-NEXT loops of lines 140, 150, and 160 to run a partial kill, or you can kill the remainder of the DOS manually. I've had a few problems with disks that had directory errors and did not complete the kill.

The program, as listed, will kill the DOS on TRSDOS 2.1, 2.2 and 2.3. If you want to use it to kill NEWDOS or NEWDOS+ (except NEWDOS 80), make minor changes as shown in Listing 2.

For NEWDOS 80, make the changes shown in Listing 3.■

```
60   REM***THIS PGM FOR NEWDOS OR NEWDOS+ (EXCEPT 60)
140  FOR I=1 TO 13: READ A($I): NEXT I
150  DATA "SYSO/SYS","SYS2/SYS","SYS3/SYS","SYS4/SYS","SYS5/SYS",·
     "SYS6/SYS","SYS11/SYS","SYS12/SYS",SYS13/SYS","FORMAT/CMD",
     "BASIC/CMD","COPY/CMD"
160  FOR J=1 TO 13: C$="KILL  "A$(J)+":"+D$
190  PRINT: PRINT "YOU HAVE 22 MORE GRANS ON THIS DISK"
```

Listing 2. KILDOS Changes for NEWDOS and NEWDOS+

```
60   REM***THIS PGM FOR NEWDOS 60
140  FOR I=1 TO 15: READ A($I): NEXT I
150  DATA "SYSO/SYS","SYS1/SYS","SYS2/SYS","SYS3/SYS","SYS4/SYS",
     "SYS5/SYS","SYS6/SYS","SYS7/SYS"."SYS8/SYS","SYS9/SYS",
     "SYS10/SYS","SYS11/SYS","SYS12/SYS",SYS13/SYS","BASIC/CMD",
160  FOR J=1 TO 15: C$="KILL  "A$(J)+":"+D$
190  PRINT: PRINT "YOU HAVE 25 MORE GRANS ON THIS DISK"
```

Listing 3. KILDOS Changes for NEWDOS 80

PROGRAMMING TOOLS FOR YOUR TRS-80™

INSIDE LEVEL II

The Programmers Guide to the TRS-80 ROMS

INSIDE LEVEL II is a comprehensive reference guide to the Level II ROMs which allows the machine language or Basic programmer to easily utilize the sophisticated routines they contain. Concisely explains set-ups, calling sequences, and variable passage for number conversion, arithmetic operations, and mathematical functions, as well as keyboard, tape, and video routines. Part II presents an entirely new composite program structure which loads under the SYSTEM command and executes in both Basic and machine code with the speed and efficiency of a compiler. In addition, the 18 chapters include a large body of other information useful to the programmer including tape formats, RAM useage, relocation of Basic programs, USR call expansion, creating SYSTEM tapes of your own programs, interfacing of Basic variables directly with machine code, a method of greatly increasing the speed at which data elements are stored on tape, and special precautions for disk systems. INSIDE LEVEL II is a clearly organized reference manual. It is fully typeset and packed with nothing but useful information. It does not contain questions and answers, ROM dumps, or cartoons. **INSIDE LEVEL II.....$15.95**

SINGLE STEP THROUGH RAM OR ROM

STEP80 allows you to step through any Basic or machine language program one instruction at a time, and see the address, hexadecimal value, Zilog mnemonic, register contents, and step count for each instruction. The top 14 lines of the video screen are left unaltered so that the 'target program' may perform its display functions unobstructed. STEP80 will follow program flow right into the ROMs, and is an invaluable aid in learning how the ROM routines function. Commands include step (trace), disassemble, run in step mode at variable step rate, display or alter memory or CPU registers, jump to memory location, execute a CALL, set breakpoints in RAM or ROM, write SYSTEM tapes, and relocate to any page in RAM. The display may also be routed to your line printer through the device control block so custom print drivers are automatically supported. **STEP80.....$16.95**

TELECOMMUNICATIONS PROGRAM

This machine language program allows reliable high speed file transfers between two disk-based computers over modems or direct wire. It is menu driven and extremely simple to use. Functions include real-time terminal mode, save RAM buffer on disk, transmit disk file, receive binary files, examine and modify UART parameters, program 8 custom log-on messages, automatic 16-bit checksum verification of accurate transmission and reception, and many more user conveniences. Supports line printers and lowercase characters. With this program you will no longer need to convert machine language programs to ASCII for transmission, and you will know immediately if the transmission was accurate. **TELCOM.....$29.95**

PROGRAM INDEX FOR DISK BASIC

Assemble an alphabetized index of your entire program library from disk directories. Program names and free space are read automatically (need not be typed in) and may be alphabetized with a fast Shell/Metzner sort by disk or program. The list may also be searched for any disk, program, or extension; disks or programs added or deleted, and the whole list or any part sent to the printer. Finally, the list itself may be stored on disk for future access and update. 'The best thing since sliced bread' (January issue of '80 Microcomputing). Works with TRSDOS, NEWDOS, and NEWDOS/80. One drive and 32K required. **INDEX.....$19.95**

4 SPEED OPTIONS FOR YOUR TRS-80

The SK-2 clock modification allows CPU speeds to be switched between normal, an increase of 50%, or a 50% reduction; selectable at any time without interrupting execution or crashing the program. Instructions are also given for a 100% increase to 3.54 MHz, though the TRS-80 is not reliable at this speed. The SK-2 may be configured by the user to change speed with a toggle switch or on software command. It will automatically return to normal speed any time a disk is active, requires no change to the operating system, and has provisions for adding an LED to indicate when the computer is not at normal speed. It mounts inside the keyboard unit with only 4 necessary connections for the switch option (switch not included), and is easily removed if the computer ever needs service. The SK-2 comes fully assembled with socketed IC's and illustrated instructions. **SK-2.....$24.95**

INSTANT ASSEMBLER

The **INSTANT ASSEMBLER** is a new, powerful tape-based editor/assembler and debugger for the TRS-80 Model I. It features immediate detection of errors as the source code is entered, assembly to memory as well as to tape, a built-in single-stepping debugger, a compactly coded source format that uses 1/3 as much memory as standard source, the ability to produce relocatable code modules, and the ability to link-load independently written modules. In addition, the INSTANT ASSEMBLER has many operational features including single stroke entry of DEFB and DEFW, continuous editing of succesive lines, alphabetic listing of symbol table, separate commands for listing error lines or the symbol table, block move function, and verification of source tapes.
INSTANT ASSEMBLER includes three separate programs. The assembler itself includes the single-stepper and debugger. In this mode you may have full register displays, decimal or hex entry, forward or backward memory displays, disassembly of object code in memory, memory display in ASCII format, and hex-to-decimal or decimal-to-hex conversion. The single-stepper will step one instruction at a time or at a fast rate to any defined address. During assembly you may quickly switch from assembler to debugger and back again without losing the source code. This makes INSTANT ASSEMBLER an excellent learning tool for machine language programming. Also included on the tape are two versions of the linking loader which allow you to write your programs in smaller modules and link them together for final assembly.
INSTANT ASSEMBLER occupies 8375 bytes of memory. In a 16K machine this will leave you more than 7000 bytes which is enough to write assembly language programs of around 2000 bytes. This makes it ideal for users with only 16K machines. While this version was written specifically for tape systems, we will soon have a disk version as well. The instruction manual may be purchased separately for $5, which will apply towards the purchase of the INSTANT ASSEMBLER. **INTASM.....$29.95**

RAM SPOOLER AND PRINT FORMATTER

This program is a full feature print formatting package featuring user defineable line and page length (with line feeds inserted between words or after punctuation), screen dump, printer pause control, and baud rate selection. In addition, printing is done from a 4K expandable buffer area so that the LPRINT or LLIST command returns control to the user while printing is being done. Ideal for Selectric or other slow printers. Allows printing and processing to run concurrently. Output may be directed to either the parallel port, serial port, or the video screen. **SPOOLER.....$16.95**

MACHINE CODE FAST FOURIER TRANSFORM

This complete package includes 3 versions of the machine language FFTASM routine assembled for 16, 32, and 48K machines, a short sample Basic program to access them, a 10K Basic program which includes sophisticated interactive graphing and data manipulation, and a manual of instructions and examples. The machine language subroutines use variables defined by a supporting Basic program to make data entry and retrieval extremely fast and easy for custom implementation. They perform 20 to 40 times faster than their Basic equivalent (256 points in 12.5 seconds), and require less than 1550 bytes of memory. The FFT is useful in analyzing stock market and comodity trends as well as for scientific information. **FFTASM.....$49.95**

DUPLICATE SYSTEM TAPES WITH CLONE

Make duplicate copies of any tape written for Level II. They may be SYSTEM tapes or data lists. The file name, load address, entry point, and every byte (in ASCII format) are displayed on the video screen. **CLONE.....$16.95**

RAMTEST FOR LEVEL II

This machine language program is a very thorough test for several types of RAM errors. A complete test of each individual bit in a 48K machine takes just 14 seconds. Includes a separate test for power line glitches. **RAMTEST.....$9.95**

EDIT BASIC PROGRAMS WITH ELECTRIC PENCIL

Load Basic programs or any other ASCII data file into the disk version of Electric Pencil for editing. One command from DOS quickly modifies existing files to Pencil format. One disk and 32K required. **PENPATCH.....$9.95**

MUMFORD MICRO SYSTEMS

ORDERING: Complete satisfaction is guaranteed [illegible]

Box 400-E Summerland, California 93067 (805) 969-4557

A seven program investment analysis package that will leave you bullish on microcomputers.

The Software Broker

John Harper
R #1 Box 252
Lawrenceburg, IN 47025

If you've ever dreamed of making a killing in the stock market, these programs may be of interest to you. They won't make a buy or sell decision for you, but they will help you predict future stock prices and market direction. I've included traditional guidelines for interpreting the results of each program.

Warning—The following programs may be hazardous to your financial well being! No one has ever been able to accurately predict the stock market. When using these programs as an investment aid, test and retest each to become familiar with the advantages and limitations of each. Remember—the final decision is yours.

Setting Up

You'll find most of the information you need in one of the major daily newspapers. The following figures must be tracked daily:

- Closing prices for each stock (or any other market index you select)
- Dow Jones Industrial Average
- Market Volume (transactions)
- Number of issues advancing (advances)
- Number of issues declining (declines)

Program Listing 1, UPDATE/DTA, creates and maintains your data files. These are compatible with any program requiring data from disk.

Program Listing 1 User Notes

- Stock price data must be entered in decimal form; a price of five and one half must be entered as 5.5.
- Zero must always be the last item entered, signifying that all data has been entered and identifying the end of the file.
- Stock price files may have any name.
- The Dow Jones Industrial Average must be named DOW/AVE.

```
10 CLS
20 DIM D(900)
30 INPUT "DO YOU WISH TO UPDATE AN EXISTING FILE  (U)
          OR CREATE A NEW ONE (N)";A$
40 IF A$="N" GOTO 300
50 CLS
60 PRINT "PLACE DISK WITH DATA FILE IN DRIVE"
70 INPUT "ENTER NAME OF OLD DATA FILE";B$
80 PRINT "NEW FILE WILL HAVE THE NAME   ";B$
90 INPUT "PRESS ENTER TO BEGIN.";A$
100 OPEN"I",1,B$
110 CLS
120 A=1
130 INPUT#1,D(A)
140 IF D(A)=0 GOTO 160
150 A=A+1:GOTO 130
160 CLOSE 1:INPUT"PRESS ENTER TO CONTINUE";A$
170 CLS:PRINT"ALL DATA READ FROM DISK."
180 PRINT"YOU CAN NOW ENTER THE NEW DATA "
190 PRINT
200 PRINT"ENTER ITEM #  ";A
210 INPUT D(A)
220 IF D(A)=0 GOTO 240
230 A=A+1:GOTO  200
240 CLS:PRINT" ALL DATA HAS BEEN ENTERED."
250 PRINT"PLACE DISK TO CONTAIN THE UPDATED DATA IN DRI
    VE"
260 INPUT"PRESS ENTER TO CONTINUE.";A$
270 OPEN"O",1,B$
280 FOR J=1TOA:PRINT#1,D(J):NEXT
290 CLOSE:END
300 CLS
310 INPUT"ENTER NAME OF FILE TO CONTAIN DATA";B$
320 CLS
330 A=1
340 PRINT "ENTER ITEM # ";A
350 INPUT D(A)
360 IF D(A)=0 GOTO 240
370 A=A+1:GOTO 340
380 END
```

Program Listing 1. UPDATE/DTA, used to create and maintain data files used by the market analysis programs.

"Warning—The following programs may be hazardous for your financial well being!"

- The file containing daily advances must be named ADVANCES.
- The file containing daily declines must be named DECLINES.
- The file containing daily volume must be named VOLUME.

Step 1: PLACE STOCK PRICE DISK IN DRIVE
ENTER NAME OF DATA FILE?
LIONEL/P
(Note: Computer instructions appear in capitals. Your response is underlined.)

Step 2: PLACE INDEX VALUE DISK IN DRIVE
ENTER NAME OF DATA FILE?
DOW/AVE

Step 3: Computer plots historical Dow Jones Industrial Average to video. When graph is complete, you must press any key to continue.

Step 4: DO YOU WANT OUTPUT TO LINE PRINTER (Y/N)? ____
YES: Bar graph of Dow Jones Index output to line printer.
NO: Program advances to next step.

Step 5: Computer plots historical stock prices on video. Press any key to continue.

Step 6: DO YOU WANT OUTPUT TO LINE PRINTER (Y/N)? ____
Same as Step 4 except output for stock prices.

Step 7: Computer extends stock price and index into the future using least-squares regression analysis. Stock prices and index values are extended independent of one another, as if they will continue their past trend.

Step 8:. Computer plots projected Dow Jones Average to the video screen. This is a plot of the least squares line of best fit which fits the data calculated in Step 7. Press any key to continue.

Step 9: Computer plots projected stock prices, same as Step 8 except for stock prices. Press any key to continue.

Step 10: DO YOU WANT OUTPUT TO LINE PRINTER (Y/N)? ____
YES (Y): Computer plots projected Dow Jones Industrial Averages line of best fit and stock price projected line of best fit on the same graph. This gives the investor an excellent view of how his stock is performing relative to the market index. It also shows direction and rate of change of stock price and market.

Step 11: Computer does regression analysis correlating historical stock prices and index values to each other. This provides a statistical measure of the relationship between price and Index.
ENTER STOCK PRICE YOU WISH TO OBTAIN? (enter any price)
THE INDEX MUST REACH —835.20— (835.20 is calculated by the computer. This value will vary depending on the stock price you wish to obtain. This assumes the past relationship between price and Index continues into the future.
THE CORRELATION COEFFICIENT —.6938— (.6938 is calculated by the computer; it is the measure of the relationship between price and index. The closer this value is to 1, the higher the degree of relationship.

FORECAST/MKT Sample Execution

Program Listing 2. FORECAST/MKT develops a quantitative measure of the historical market and stock price trend.

```
10 CLS
20 DIM P(800),V(800),Q(800),I1(800),V1(800),I2(400),V2(
   400)
30 REM * THIS PROGRAM USES HISTORICAL STOCK PRICES AND"

40 REM * INDEX VALUES TO PROJECT FUTURE PRICES *
50 GOSUB 940
60 REM * THIS SECTION INPUTS INDEX VALUES *
70 CLS
80 GOSUB 800
90 NZ=N
100 IF A<>N GOSUB 2890
110 PRINT"THE COMPUTER WILL NOW GRAPH THE HISTORICAL PR
    ICE"
120 PRINT",INDEX, AND THE PROJECTED  PRICE AND INDEX."
130 PRINT"AFTER EACH GRAPH IS DRAWN, PRESS ANY KEY TO"
140 PRINT"CONTINUE."
150 U3$="D J INDUSTRIALS"
160 GOSUB 1670
170   GOSUB 2210
180 U3$="STOCK PRICES"
190 GOSUB 1720
200  GOSUB 2280
210 GOSUB 1070
220  GOSUB 2550
230 REM * THIS SECTION COMPUTES THE SLOPE AND Y INTERCE
    PT *
240 REM * OF THE LINE OF BEST FIT FOR THE DATA *
250 CLS
260 FOR A=1TON-1
270 A1=V(A)+A1
280 A2=P(A)+A2
290 B1=V(A)*P(A)+B1
300 B2=P(A)[2+B2
310 NEXT:A1=A1/(N-1):A2=A2/(N-1)
320 FORA=1TON-1
330 A5 = ((V(A)-A1)[2)+A5
340 A3=((V(A)-A1)*(P(A)-A2))+A3
350 NEXT
360 A4=A3/(N-1)
370 A6=A5/(N-1)
380 SL=(B1-((N-1)*A1*A2))/(B2-((N-1)*(A2[2)))
390 IC=A1-(SL*A2)
400 REM * THIS SECTION COMPUTES THE CORELATION COEFFICI
    ENT *
410 PA=A2
420 IA=A1
430 B1=0:B2=0:B3=0
440 FOR A=1 TO N-1
450 B1=((P(A)-PA)*(V(A)-IA))+B1
460 B2=((P(A)-PA)[2)+B2
470 B3=((V(A)-IA)[2)+B3
480 NEXT
490 R=(1/(N-1)*B1)/(SQR((1/(N-1))*B2)*SQR((1/(N-1))*B3)
    )
500 PRINT
510 PRINT "THE EQUATION OF THE LINE OF BEST FIT IS : "
520 PRINT
530 PRINT      "Y = ";IC;" + ";SL;" * X "
540 PRINT
550 PRINT "YOU MAY NOW ENTER THE STOCK PRICE YOU WISH T
    O "
560 PRINT "OBTAIN.  THE COMPUTER WILL RESPOND BY GIVING
     YOU"
570 PRINT "THE LEVEL YOUR PRICE INDEX MUST REACH BEFORE
     YOU "
580 PRINT "ACHEVE THIS STOCK PRICE.  THIS WILL BE FOLLO
    WERED "
590 PRINT "BY THE COORELATION COEFFICIENT--A MEASURE OF
     THE "
600 PRINT "DEGREE OF RELATIONSHIP BETWEEN STOCK PRICES
    AND "
610 PRINT "INDEX VALUES."
620 PRINT:INPUT "PRESS ENTER TO CONTINUE";A$
630 CLS
640 INPUT "ENTER STOCK PRICE YOU WISH TO OBTAIN";PR
650 IV=IC+(SL*PR)
660 CLS
670 PRINT"THE INDEX MUST REACH ";IV
680 PRINT"THE COORELATION COEFFICIENT = ";R
690 PRINT"YOUR STOCK PRICE IS ";PR
700 IF W$="N" GOTO 740
710 LPRINT "TRIAL PRICE    =     ";PR
720 LPRINT "NECESSARY INDEX    =   ";IV
730 LPRINT
740 PRINT"TO RUN AGAIN TYPE 1"
750 PRINT"TO CONTINUE TYPE 2"
```

Program continues

```
760 INPUT Z
770 IF Z=1 GOTO 630
780 CLS
790 END
800 REM * READS INDEX VALUES FROM TAPE *
810 CLS
820 PRINT"PLACE INDEX VALUE DISK IN DRIVE"
830 INPUT "ENTER NAME OF DATA FILE.";XV$
840 A=1
850 OPEN"I",2,XV$
860 INPUT#2,V(A)
870  IF V(A)>0 GOTO 890
880 GOTO 920
890 V$=INKEY$:IF V$="S" GOTO920
900 A=A+1
910 GOTO 860
920 CLOSE 2
930 RETURN
940 CLS
950 PRINT"PLACE STOCK PRICE DISK IN DRIVE"
960 INPUT"ENTER NAME OF DATA FILE";XZ$
970 A=1:N=1
980 OPEN"I",1,XZ$
990 INPUT#1,P(A)
1000 IF P(A)>0  GOTO  1020
1010 GOTO 1050
1020 V$=INKEY$:IF V$="S" GOTO 1050
1030 A=A+1:N=N+1
1040 GOTO 990
1050 CLOSE 1
1060 RETURN
1070 CLS
1080 E1=0:F1=0:F2=0:E2=0:E3=0:F3=0
1090 FOR A=1 TO N-1
1100 E1=V(A)*A+E1
1110 F1=V(A)+FI
1120 F2=A+F2
1130 E2=A[2+E2
1140 E3=P(A)*A+E3
1150 F3=P(A)+F3
1160  NEXT
1170 F1=F1/(N-1):F2=F2/(N-1):F3=F3/(N-1)
1180  S2=(E1-((N-1)*F1*F2)):S7=(E2-((N-1)*(F2[2)))
1190 S2=S2/S7
1200 I2=F1-(S2*F2)
1210 S3=(E3-((N-1)*F3*F2))/(E2-((N-1)*(F2[2)))
1220  I3=F3-(S3*F2)
1230 GOTO 1310
1240 LPRINT "E1= ";E1
1250  LPRINT"F1= ";F1;"F2= ";F2
1260 LPRINT"E3= ";E3;"  F3= ";F3
1270 LPRINT "S2= ";S2
1280 LPRINT "I2= ";I2
1290 LPRINT "S3= ";S3
1300 LPRINT  "I3= ";I3
1310 PRINT "THE COMPUTER WILL NOW EXTEND THE STOCK PRIC
     E"
1320 PRINT "AND INDEX VALUE ";(N-1)/2;" DAYS INTO THE F
     UTURE."
1330 PRINT "DAY ONE WILL BE TOMOROWS PROJECTED PRICE/IN
     DEX"
1340 PRINT:INPUT"PRESS ENTER TO BEGIN.";A$
1350 CLS
1360 INPUT"DO YOU WANT OUTPUT TO LINE PRINTER (Y/N)";W$

1370 A9=1
1380 Z9=1
1390 IF W$="N" GOTO 1430
1400 LPRINT CHR$(15)"PROJECTED":
1410 LPRINT"PRICE               INDEX"
1420 FOR P39=1 TO 80:LPRINT"*";:NEXT
1430  FOR A=N-1 TO (((N-1)/2)+(N-1))
1440 Y2=I2+(S2*A)
1450 Y3=I3+(S3*A)
1460 V1(Z9)=Y2:V2(Z9)=Y3:I1(Z9)=Y2:I2(Z9)=Y3
1470 Z9=Z9+1
1480 PRINT "DAY ";A9;TAB(20)"STOCK PRICE";Y3;TAB(50)"IN
     DEX";Y2
1490 IF W$="Y" LPRINT Y3,Y2
1500 A9=A9+1
1510 FOR Z=1 TO 500:NEXT Z
1520 NEXT A
1530 IF W$="Y" LPRINT:LPRINT
1540 INPUT "PRESS ENTER TO CONTINUE";A$
1550 NZ=N/2
1560 KN=INT(N-1)/2:LL=90
1570 U3$="D J INDUS. PROJECTION"
1580 GOTO 1770
1590  FOR A=1 TO KN
1600 I1(A)=I2(A)
1610 NEXT
```

Program continues

The past is one thing we can be sure of—it will never change. Program Listing 2, FORECAST/MKT, measures market history and stock price trends. It describes graphically the past action of the market. It measures a selected stock against the Dow Jones Industrial Average. (The Dow Jones Industrial Average is a market index that measures the general economic health of the country's business.)

Let's take our first look at the unknown—the future. Fit a straight line to the past data (least squares line of best fit). Extend this line into the future, as if the past trend of the stock and market will continue unchanged. This produces our first look at where the past wants to take us; what the future actually brings may be shockingly different.

Interpret the results of this program with extreme caution. The program projects stock and market values into the future as*suming they continue their past performance and relationship to each other.* Needless to say, this can't be guaranteed.

Line Number	Function
50	Enters stock prices from disk.
80	Enters index values from disk.
100	There must be an equal number of stock prices and Index values, otherwise this line creates two arrays of equal length.
160	Plots Dow Jones Index to video screen.
170	Optional output to line printer.
190	Plots stock prices to video screen.
200	Optional output to line printer.
210	Extends stock prices and index into the future assuming they maintain their past performance.
220	Optional output to line printer.
400	Computes coefficient of correlation that exists between stock prices and index values.
550	Using the correlation between price and index, the computer describes the relationship between them. At this point enter the stock price you wish to obtain. The computer will respond with the level the Dow Jones Average will probably have to reach before your stock reaches your desired price.

Table 1. FORECAST/MKT Program Flow

Details of the program are shown in Table 1. To run it, the operator must have the stock price and market index data stored on disk. In the following example, stock prices are stored under the name LIONEL/P and the index is stored as DOW/AVE.

Limits of the Past

When we extend our line into the future, we assume the past trend will continue unchanged. This is a very dangerous, and wrong, assumption. At some point in time the past trend will change.

We must decide when this change will occur and what it will look like. From this point on, we are walking on air. Fortunately,

"The past is one thing we can be sure of—it will never change."

```
1620 U3$="STOCK PRICE PROJECTION"
1630 LL=95:CLS:GOTO 1770
1640 RETURN
1650 END
1660 END
1670 FOR Z=1 TO N-1
1680 I1(Z)=V(Z)
1690 NEXT
1700 KN=N-1:KL=(KN/11)
1710 GOTO 1770
1720 FOR Z=1 TO N-1
1730 I1(Z)=P(Z)
1740 NEXT
1750 KN=N-1:KL=INT((KN/11)+.9)
1760 GOTO 1770
1770 P2=0:P=0
1780 REM * SEARCH FOR HIGH AND LOW *
1790 IL=2000:IH=0
1800 FOR Z=1 TO KN
1810 IF I1(Z)<IL THEN IL = I1(Z)
1820 IF I1(Z)>IH THEN IH=I1(Z)
1830 NEXT
1840 RI=IH-IL:RI=(RI/6)
1850 CLS
1860 PRINT @ 25,U3$
1870 P2=0
1880 FOR X=IH TO IL STEP -RI:PRINT @(P2),X;:PRINT@(P2+5
     ),"     "::P2=P2+128:NEXT
1890 FOR X=9 TO 127:SET(X,40):NEXT
1900 FOR X=12 TO 127 STEP 10:SET(X,39):NEXT
1910 PRINT @(900),"1";:PRINT TAB(20),"<--DAYS-->";:PRIN
     T @(951),NZ:
1920 FOR Y=0 TO 39:SET(11,Y):NEXT
1930 FOR Y=3 TO 39 STEP 2:SET(10,Y):NEXT
1940 SR=INT((KN/115)+.9):IF KN<100 SR=SR/(100/KN)
1950 IF KN<11 PRINT"YOU MUST HAVE AT LEAST 11 ELEMENTS
     TO CONSTRUCT GRAPHS":GOTO 2180
1960 X1=13:X2=IH-IL:X2=X2/40
1970 F2=1
1980 FOR X=1 TO KN STEP SR
1990 X3=40-((I1(X)-IL)/X2)
2000 SET(X1,X3)
2010 Q(F2)=X3
2020 X1=X1+1:F2=F2+1
2030 IF X1>127 GOTO 2050
2040 NEXT
2050 X1=13:F2=1
2060 FOR X=1 TO (KN-SR)STEP SR
2070 IF Q(F2)<Q(F2+1) GOTO 2130
2080 FOR Z=INT(Q(F2)) TO INT(Q(F2+1)) STEP -1
2090 IF X1>127 GOTO 2170
2100 SET(X1,Z)
2110 NEXT Z
2120 GOTO 2160
2130 FOR Z=Q(F2) TO Q(F2+1)
2140 SET(X1,Z)
2150 NEXT Z
2160 X1=X1+1:F2=F2+1:NEXT X
2170 B$=INKEY$:IF B$="" GOTO 2170
2180 IF LL=90 GOTO 1590
2190 IF LL=95 GOTO 1640
2200 RETURN
2210 REM * OUTPUT TO QUICK PRINTER *
2220 INPUT "DO YOU WANT OUTPUT TO LINE PRINTER (Y/N)";W
     $
2230 IF W$="N" THEN RETURN
2240 FOR A=1 TO N
2250 V1(A)=V(A)
2260 NEXT
2270  GOTO 2360
2280 REM  * OUTPUT TO QUICK PRINTER *
2290 INPUT"DO YOU WANT OUTPUT TO LINE PRINTER (Y/N)";W$
2300 IF W$="N" THEN RETURN
2310  LPRINT:LPRINT
2320 FOR A=1 TO N
2330 V1(A)=P(A)
2340 NEXT
2350 GOTO 2360
2360  H=0:L=V1(1)
2370 FOR X=1 TO N-1
2380 IF V1(X)>H THEN H=V1(X)
2390 IF V1(X)<L THEN L=V1(X)
2400 NEXT
2410 LPRINT"     ";U3$
2420 LPRINT L;TAB(30)"<--RANGE-->";TAB(73)H
2430 FOR P39=1 TO 75:LPRINT"-";:NEXT:LPRINT" "CHR$(13)
2440 RA=H-L:RA=78/RA
2450 FOR X=1 TO N-1
2460 LPRINT":";
2470 T=RA*(V1(X)-L):T=INT(T)
2480 FOR Z=1 TO T
```

Program continues

"One point to keep in mind before going on—these programs attempt to measure trend and not stock price trend."

```
2490 LPRINT TAB(Z)"*";
2500 NEXT  Z
2510 LPRINT
2520 NEXT X
2530 LPRINT
2540 RETURN
2550 A8=INT((N-1)/2)
2560 INPUT"DO YOU WANT OUTPUT TO LINE PRINTER (Y/N)";W$

2570 IF W$="N"  THEN RETURN
2580 H=0:H1=0:L=2000:L1=2000
2590 H1=0
2600 FOR X=1 TO A8
2610 IF V1(X)>H THEN H=V1(X)
2620 IF V2(X)>H1 THEN H1=V2(X)
2630 IF V1(X)<L THEN L=V1(X)
2640 IF V2(X)<L1 THEN L1=V2(X)
2650 NEXT
2660 LPRINT"S = PRICE AND INDEX PLOT AT SAME LOCATION"
2670 LPRINT:LPRINT"I = INDEX  ----  P = STOCK PRICE"
2680 LPRINT CHR$(15);"PRICE VS INDEX"
2690 FOR P39=1 TO 75:LPRINT"-";:NEXT:LPRINT" ";CHR$(13)

2700 R1=H-L:R1=60/R1
2710 R2=H1-L1:R2=60/R2
2720 FOR X=1 TO A8
2730 LPRINT":";
2740 T1=R1*(V1(X)-L):T1=INT(T1)
2750 T2=R2*(V2(X)-L1):T2=INT(T2)
2760 PRINT V1(X),V2(X)
2770 IF T1=T2 GOTO 2830
2780 IF T1<T2 GOTO 2810
2790 LPRINT TAB(T2)"P";TAB(T1)"I"
2800 GOTO 2840
2810 LPRINT TAB(T1)"I";TAB(T2)"P"
2820 GOTO 2840
2830 LPRINT TAB(T1)"S"
2840 NEXT X
2850 LPRINT
2860 RETURN
2870 END
2880 END
2890 CLS
2900 PRINT A;" INDEX VALUES ENTERED"
2910 PRINT N;" STOCK PRICES ENTERED"
2920 PRINT "THIS SECTION CHOPS OFF THE FRONT END OF THE
     "
2930 PRINT "LARGEST ARRAY (STOCK OR INDEX) RESULTING IN
     "
2940 PRINT "TWO ARRAYS OF EQUAL LENGTH.  BOTH ARRAYS MU
     ST"
2950 PRINT "BE OF EQUAL LENGTH BEFORE PROGRAM EXECUTION
     ."
2960 IF A>N GOTO 3050
2970 QA=N-A
2980 BM=1
2990 FOR X=QA+1 TO N
3000 P(BM)=P(X)
3010 BM=BM+1
3020 NEXT
3030 N=A
3040 RETURN
3050 QA=A-N
3060 BM=1
3070 FOR X=QA+1 TO A
3080 V(BM)=V(X)
3090 BM=BM+1
3100 NEXT
3110 RETURN
3120 END
3130 A=N
```

we have a parachute; unfortunately, we do not know if it will open.

The remaining programs try to predict the change in trend and measure the risk involved in a particular stock. Each have their own advantages and disadvantages. At times the programs will give conflicting results and you must decide which results are more reliable.

Potential Reward... Guaranteed Risk

Program Listing 3, STOCK/ANA, estimates the risk and rate of return associated with a particular stock. The analysis is based on the historic relationship between stock price and index values and the past earnings per share of the stock. Remember that any future change in these relationships can change the amount of risk associated with that particular stock.

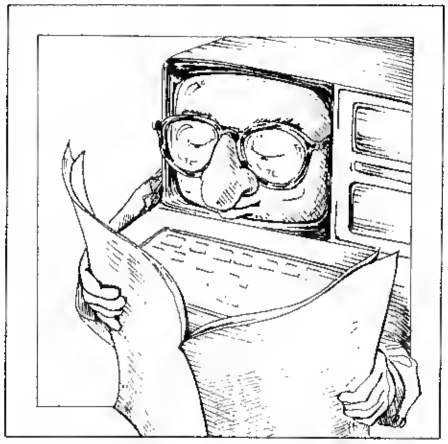
The program uses three statistical measures to estimate risk and return:

Rate of change in Earnings Per Share (EPS): As a general rule, the greater the variation in EPS from year to year, the greater is the uncertainty of future price. When measuring variation, look at *rate* of change and not the *amount* of change. To calculate rate of change, apply the least squares equation to the common logarithms of EPS, then calculate trend values in logarithms and convert those values back to dollar values by taking anti-logs of the logarithms. (See formula, Table 3.)

The Beta Coefficient is a quantitative comparison of the price of a stock in relation to an index (DJI) for the same time period. A Beta that is greater than one (1.00) normally indicates your stock is less stable than the market. If the market increases, your stock would be expected to increase faster than the market.

If the market decreases, your stock would be expected to decrease at a greater rate than the market. As an investor, you would want to hold stocks with high Betas if you expect the market to rise.

A Beta that is less than one would be expected to be more stable than the market. In a rising market, a low Beta stock would not rise as much as the market. In a falling market, a low Beta stock would not be expected to fall as much as the market. If the market is expected to fall, an investor would want a portfolio of low Beta stocks, providing a degree of protection during a market decline.



Interpretation of Beta Coefficient:
Beta = 1.08

Rising Market. In a rising market the stock would be expected to increase 8 percent more than the market.

Falling Market. In a falling market the stock would be expected to decline 8 percent more than the market.

Beta = .95

Rising Market. In a rising market the stock would be expected to increase 5 percent less than the market.

Falling Market. In a falling market the

"This produces our first look at where the past wants to take us; what the future actually brings may be shockingly different."

stock would be expected to decline 5 percent less than the market.

The Alpha Value is a measure of the rate of return of an individual stock relative to the rate of return from the market index. An Alpha Value greater than zero indicates the rate of return from the stock was greater than the return from the index. An Alpha less than zero indicates the rate of return was less than the index rate of return.

How effectively does the Beta Coefficient measure your risk? Until recently it was widely accepted as *the* measure of risk. It still retains a sizable following, but has come under attack. One major criticism of Beta is its use of the Dow Jones Industrial Average as a measure of the overall market performance. Large organizations today use computers to measure the entire market, rather than trying to estimate it with a market average. As this is a bit outside the ability of the microcomputer, you must decide for yourself how trustworthy the Beta is.

When selecting stock there is always a degree of risk. As a general rule, the greater the risk, the greater the potential profit. Each investor must decide for himself how much risk he is willing to accept. Never accept more than you feel comfortable with; never gamble unless you can afford to lose.

STOCK/ANA: Sample Execution

Step 1: PLACE STOCK PRICE DISK IN DRIVE
ENTER NAME OF DATA FILE? LIONEL/P

Step 2: PLACE DISK CONTAINING INDEX DATA IN DRIVE ENTER NAME OF DATA FILE? DOW/AVE

Step 3: Computes standard deviation and variance:

STANDARD DEVIATION OF PRICE	.044
STANDARD DEVIATION OF INDEX	1.45
VARIANCE OF INDEX VALUES	2.10
VARIANCE OF STOCK PRICES	1.9545E-03

Step 4: Computes Alpha and Beta coefficient:

ENTER DIVIDEND FOR LAST QUARTER? .30

BETA COEFFICIENT	1.0524
ALPHA VALUE	−5.0948

Step 5: Computes rate of change in earnings per share (EPS):

ENTER NUMBER OF YEARS FOR WHICH EARNINGS PER SHARE (EPS) IS KNOWN? 12

ENTER EPS FOR YEAR 1	.63
ENTER EPS FOR YEAR 2	.12
ENTER EPS FOR YEAR 3	.25
ENTER EPS FOR YEAR 4	.08
ENTER EPS FOR YEAR 5	.23
ENTER EPS FOR YEAR 6	.33
ENTER EPS FOR YEAR 7	.34
ENTER EPS FOR YEAR 8	.17
ENTER EPS FOR YEAR 9	.30
ENTER EPS FOR YEAR 10	.39
ENTER EPS FOR YEAR 11	.68
ENTER EPS FOR YEAR 12	1.00

AVERAGE ANNUAL RATE OF CHANGE IN EPS = 10.88

Step 6: Computes coefficient of determination:

COEFFICIENT OF DETERMINATION = .242279

Program Listing 3. STOCK/ANA estimates the risk and rate of return associated with a particular stock, based on the historic relationship between stock price and index values and the past earnings per share of the stock.

```
10 CLS
20 DIM P(800),V(800),I(800),E(20),N3(20),LY(20),Y(20),I
   P(800)
30 GOSUB 1630
40 REM * THIS SECTION INPUTS INDEX VALUES *
50 CLS
60 GOSUB 1380
70 NZ=N
80 REM * THIS SECTION COMPUTES STANDARD DEVIATION AND V
   ARIANCE
90  PZ=0:PW=0
100 FOR A=1TON-1
110 PZ=P(A)+PZ:PW=V(A)+PW
120 NEXT
130  PA=PZ:IA=PW
140 PZ=PZ/(N-1):PW=PW/(N-1)
150 SP=0:SI=0
160 FOR A=1 TO N-1
170 SP=((P(A)-PZ)[2)+SP
180 SI=((V(A)-PW)|2)+SI
190 NEXT
200 SP=SQR(SP)/(N-1)
210 SI=SQR(SI)/(N-1)
220 VP=SP[2
230 VI=SI[2
240 CV=SP/PA
250 CI=SI/IA
260 CLS:PRINT:PRINT:PRINT
270 PRINT"STANDARD DEVIATION OF PRICE   ";SP
280 PRINT"STANDARD DEVIATION OF INDEX   ";SI
290 PRINT"VARIANCE OF INDEX VALUES      ";VI
300 PRINT"VARIANCE OF STOCK PRICES      ";VP
310 PRINT:PRINT
320 INPUT "PRESS ENTER TO CONTINUE";A$
330 REM * COMPUTES BETA COEFFICIENT *
340 CLS
350 INPUT "ENTER DIVIDEND FOR LAST QUARTER ";C2
360 E1=0:E2=0
370 D=C2/90
380 FORA=1TON-2
390 I(A)=((P(A+1)+D)/P(A))*100
400 IP(A)=(V(A+1)/V(A))*100
410 NEXT
420 MP=0:MI=0
430 FOR A=1TON-2
440 MP=IP(A)+MP
450 MI=I(A)+MI
460 NEXT
470 MP=MP/(N-2)
480 MI=MI/(N-2)
490 FOR A=1TON-2
500 E1=(IP(A)-MP)*(I(A)-MI)+E1
510 E2=((IP(A)-MP)[2)+E2
520 NEXT
530 E3=MI-((E1/E2)*MP)
540 E4=E1/E2
550 CLS
560 PRINT:PRINT
570 PRINT "BETA COEFFICIENT = ";E4
580 PRINT"****************************************"
590 PRINT:PRINT "ALPHA VALUE ";TAB(17)"= ";E3
600 PRINT:PRINT"****************************************"
610 PRINT
620 IF W$="Y" LPRINT "ALPHA = ";E3
630 IF W$="Y" LPRINT "BETA  = ";E4:LPRINT:LPRINT
640 INPUT"PRESS ENTER TO CONTINUE";A$
650 REM * COMPUTES VARIATION EN EARNINGS PER SHARE (EPS
    ) *
660 CLS
670 INPUT "ENTER NUMBER OF YEARS FOR WHICH EARNINGS PER
     SHARE IS KNOWN.";N1
680 CLS
690 FOR A=1 TO N1
700 CLS
710 PRINT "ENTER EPS FOR YEAR # ";A
720 INPUT E(A)
730 NEXT
740 FOR A=1 TO N1
750 IF E(A)<0 THEN A8=ABS(E(A))+A8 ELSE A9=E(A)+A9
760 NEXT
770 IF A8>=A9 GOTO 850
780 K=0
790 FOR A=1 TO N1
800 IF(E(A)<0)*(ABS(K)<ABS(E(A)))THEN K=ABS(E(A))
810 NEXT
820 FOR A=1 TO N1
```

Program continues

COTTAGE SOFTWARE

PACKER: Automatically edits all or part of your Basic program to ease editing, run faster, or save memory. Has 5 sections: UNPACK—unpacks multiple statement lines into single statements maintaining program logic, inserts spaces and renumbers lines for easier editing. SHORT—shortens your program by editing out all REM statements, unnecessary words and spaces. PACK—executes UNPACK & SHORT then packs lines into multiple statement lines; maintains program logic. RENUM—renumbers program lines including all branches. You specify increment. MOVE—moves any line or block of lines to any new location in the program and renumbers lines. Written in machine language. Supplied on tape in 3 versions for 16K, 32K & 48K.
For TRS-80™ Level II or Disk Basic........................**$29.95**

FAST SORT ROUTINES: for use with Radio Shack's Accounts Receivable, Inventory Control I, & Disk Mailing List Systems for Model I Level II. Sorts in SECONDS! You'll be amazed at the time they can save. Supplied on data diskette with complete instructions.
FAST SORT for Accounts Receivable.................**$19.95**
FAST SORT for Inventory Control I......................**$19.95**
FAST SORT for Disk Mailing List (specify data diskette or tape for one drive system)...........................**$14.95**
ALL THREE ROUTINES.......................................**$44.95**

SYSTEM TAPE DUPLICATOR: Copy your system format tapes. Includes verify routine.
For any TRS-80™ Level II...**$14.95**

CASSETTE LABEL MAKER: A mini word processor to print cassette labels on a line printer. Includes manual and 50 peel-and-stick labels on tractor feed paper.
For TRS-80™ Level II & printer...............................**$15.95**

4116 RAM CHIPS—Tested!! Guaranteed for 1 year to the original purchaser.............................16K for **$49.95**

MANY MORE items available. Call or write for catalog. VISA & MASTERCHARGE accepted. Dealer inquiries invited. Kansas residents add 3% sales tax. Foreign orders in US currency only. ™
On line catalog on Wichita FORUM-80: 316-682-2113
Or call our 24 hour phone: 316-683-4811 or write:

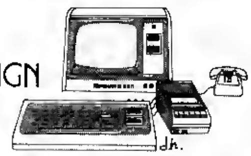
COTTAGE SOFTWARE
614 N. Harding ✓233
Wichita, KS 67208

"TRS-80 is a registered trademark of TANDY CORP."

INTERFACE EXPANSION BOARD

*For the TRS 80**

MD
MICRO-DESIGN



The Interface Expansion Board gives your computer these features:

Phone Modem
2K E-PROM OPTION
32 K Memory
Real-Time Clock
Parallel Port
RS-232 Port
Dual Cassette Line
*Floppy Disk Controller**
On Board Supply
Silk Screen
Solder Mask
Expansion Port
Manual

In Stock Now
PC Board & user manual
MDX-1
64.95
MDX-2*
74.95

Add 5.00 for shipping & handling. Texas residents add 5% sales tax.

MANUALS $7.95

FREE PAMPHLET AVAILABLE
Call or write

MICRO-DESIGN ✓379
P.O. Box 18054
Austin, Texas 78760
1-512-458-2937

**TRS 80 is a Trademark of Tandy Corp.*

OSI (8K) APPLE TRS-80†

PRESENTS: **Computers & Gambling Products Magazine***

PROBABILITY HANDICAPPING DEVICE 1 — A BASIC PROGRAM FOR: HORSE RACE HANDCAPPING!

This incredible program was written by a professional software consultant to TRW Space Systems. This is a complex program carefully human factored for easy use. It is a comprehensive horse racing system for spotting overlays in thoroughbred sprint races. Your computer will accurately predict the win probability and odds line for each horse based on your entries from the racing form. The next day overlaid horses can be spotted on the track tote board. The user's manual contains a complete explanation of overlay betting plus much more useful information. The appendix contains a detailed tab run of a 100 consecutive race system workout showing **an amazing 50% return** ($1.50 returned for each $1.00 flat wager.) Includes many features such as error correction, bubble sort, line printer output, automatic keyboard debounce, archiving, etc. The manual may be ordered separately for perusal for $7 95 and credit.

CHALLENGER 1P, 2P, or 4P 8K VERSIONS Now Available!
Phd-1 User's manual and cassette for:

Apple II (16K), TRS-80 Level II (16K), Challenger (8K)	**29.95**
TRS-80 or APPLE DISK	**34.95**

BRAND NEW FROM SDL: WIN AT THE RACES. This thoroughbred handicapping algorithm is based on a currently popular book on thoroughbred multiple regression techniques. Both sprints and routes. All of the features of PHD-1 plus more. This program incorporates the best data entry technique we've ever seen.

32K TRS-80 or APPLE CASSETTE	34.95
32K TRS-80 or APPLE DISK	39.95

BOOKS:

Winning at the Races	**21.95 + .75 P&H**
Beating the Races with a Computer	**14.95 + .75 P&H**

Make checks payable to **JOE COMPUTER DEPT. 8** ✓193
22713 Ventura Blvd., Suite F, Woodland Hills, CA 91364
CA residents add 6 % sales tax.
PHONE ORDERS: 213-992-0514
*SEND $2.00 TO PLACE YOUR NAME ON OUR MAILING LIST.
†*TRS-80 is a registered trademark of Tandy Corporation.*

DISCOVER THE 6809 IN YOUR COLOR COMPUTER

Now you can explore the Radio Shack Color Computer's impressive potentials—as an inexpensive development system, a color peripheral, a process controller—ad infinitum. The Micro Works introduces these powerful software tools for utilizing the color computer at the assembly language level.

MONITOR TAPE: A cassette tape which allows you to:

- Examine or change memory using a formatted hex display
- Save areas of memory to cassette in binary (a "CSAVEM")
- Download/upload data or programs to a host system
- Move the video display page throughout RAM
- Send or receive RS-232 at up to 9600 baud
- Investigate and activate features of your computer, such as hi-res graphics or machine-language music
- Use your color computer as an intelligent peripheral for another computer, a color display or a 6809 program development tool

The monitor has 19 commands in all, and is relocatable and re-entrant.
80C Monitor Tape Price: $29.95

MONITOR ROM: The same program as the monitor tape, supplied on ROM. This allows BASIC to use the entire RAM space. And you don't need to re-load the monitor each time you use it. The ROM plugs into the Extended Basic ROM Socket or a modified ROMPACK.
80C Monitor ROM
Price: $39.95

C BUG IS HERE!!

INSIDE THE COLOR COMPUTER: This package is a disassembler which runs on the color computer and enables you to generate your own source listing of the BASIC interpreter ROM. Also included is a documentation package which gives useful ROM entry points, complete memory map, I/O hardware details and more. Disassembler features include cross-referencing of variables and labels; output code which can be reassembled; output to an 80-column printer, small printer or screen; and a data table area specification which defaults to the table boundaries in the interpreter ROM. A 16K system is required for the use of this cassette.
80C Disassembler Price: $49.95

THE MICRO WORKS ✓109

Mastercharge and BankAmericard

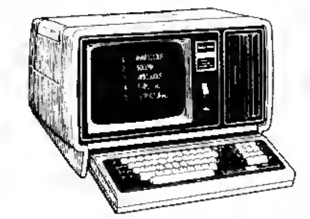
P.O. BOX 1110 DEL MAR, CA 92014 714-942-2400

"At times the programs will give conflicting results and you must decide which... are more reliable."

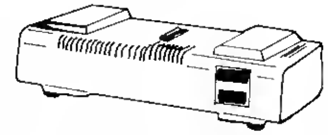
```
830 E(A)=E(A)+ABS(K)
840 NEXT
850 N2=(N1/2)-.5
860 FOR A=N1 TO 1 STEP -1
870 N3(A)=N2
880 N2=N2-1.0
890 NEXT
900 L1=0:L2=0:L3=0
910 FOR A=1 TO N1
920 L1=LOG(E(A))/LOG(10)+L1
930 L2=N3(A)*(LOG(E(A))/LOG(10))+L2
940 L3=N3(A)[2+L3
950 NEXT
960 LA=L1/N1
970 LB=L2/L3
980 FOR A=1 TO N1
990 LY(A)=LA+(LB*N3(A))
1000 NEXT
1010 FOR A=1 TO N1
1020 Y(A)=10[LY(A)
1030 NEXT
1040 SU=0
1050 FOR A=1 TO N1-1
1060 SU=((Y(A+1)/Y(A))*100)+SU
1070 NEXT
1080 VA=SU/(N1-1)
1090 VA=VA-100
1100 CLS
1110 PRINT:PRINT
1120 PRINT "AVERAGE ANNUAL RATE OF CHANGE IN EPS = ";VA
1130 PRINT"*********************************************
     "
1140 PRINT
1150 IFW$="Y"LPRINT "RATE OF CHANGE IN EPS = ";VA:LPRIN
     T:LPRINT
1160 PRINT:INPUT"PRESS ENTER TO CONTINUE";A$
1170 REM * COEFFICIENT OF DETERMINATION *
1180 M1=0:M2=0:M3=0
1190 FOR A=1 TO N1
1200 M1=E(A)+M1
1210 NEXT
1220 MA=M1/N1
1230 FOR A=1 TO N1
1240 M2=(Y(A)-MA)[2+M2
1250 M3=(E(A)-MA)[2+M3
1260 NEXT
1270 CD=M2/M3
1280 CLS
1290 PRINT:PRINT
1300 PRINT "COEFFICIENT OF DETERMINATION ";CD
1310 PRINT"************************************"
1320 PRINT
1330 IF W$="Y" LPRINT "COEFFICIENT OF DETERMINATION "
1340 IF W$="Y" LPRINT TAB(15),"  =";CD
1350 INPUT "PRESS ENTER TO CONTINUE";A$
1360 CLS
1370 END
1380 REM * READS INDEX VALUES FROM TAPE *
1390 CLS
1400 PRINT"PLACE DISK CONTAINING  INDEX DATA IN DRIVE"
1410 INPUT "ENTER NAME OF DATA FILE.";XV$
1420 A=1
1430 OPEN"I",2,XV$
1440 INPUT#2,V(A)
1450  IF V(A)>0 GOTO 1470
1460 GOTO 1500
1470 V$=INKEY$:IF V$="S" GOTO1500
1480 A=A+1
1490 GOTO 1440
1500 CLOSE 2
1510 IF A=N GOTO 1610
1520 IF N>A GOTO 1580
1530 FOR C9=1 TO N
1540 V(C9)=V(C9+(A-N))
1550 NEXT C9
1560 A=N
1570 GOTO 1610
1580 FOR C9=1 TO A
1590 P(C9)=P(C9+(N-A))
1600 NEXT:N=A
1610 PRINT
1620 RETURN
1630 CLS
1640 PRINT"PLACE STOCK PRICE DISK IN DRIVE"
1650 INPUT"ENTER NAME OF DATA FILE";XZ$
1660 A=1:N=1
1670 OPEN"I",1,XZ$
1680 INPUT#1,P(A)
1690 IF P(A)>0  GOTO  1710
1700 GOTO 1740
```

Program continues

meet a REAL PLUS

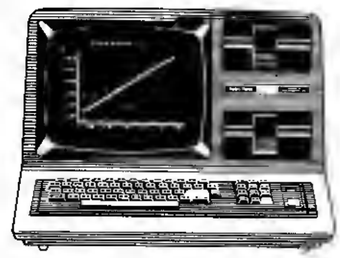


Model II 64K $3375



32K Exp. Int. $362

Color Computer 4K $310



Model III 16K $859

check out these unusual package values for price and warranty

Here are just a few of our fine offers . . . call toll-free for full information.

COMPUTERS

Model II 64K	$3375
Model III 4K LEV I	599
Model III 16K	859
Model III 32K	981.50
+ Model III 32K	915.50
Model III 48K	1104
+ Model III 48K	972
Model III 32K 2 Disc & RS232 c	2149
Color Computer 4K	310
Color Computer 16K	439.95
+ Color Computer 16K	366.50
Color Computer 16K w/extended basic	489
Pocket Computer	199
VIDEOTEX	329
APPLE 48K only	1279
ATARI 800 16K	789

+ Computer Plus New Equipment.
180 Day Extended Warranty

PERIPHERALS

Expansion Interface 0K	$249
Expansion Interface 16K	359
+ Expansion Interface 16K	305.50
Expansion Interface 32K	469.95
+ Expansion Interface 32K	362
16K RAM N.E.C. 200 N.S. chips	39

MODEMS

Lynx Direct Connect	219
COMM 80 Interface	159.95
Chatterbox Interface	239
Telephone Interface II	169

PRINTERS

Line Printer IV	849
Daisy Wheel II	1695
Line Printer VI	999
NEC Spinwriter 5530	2595
OKI DATA Microline 80	499
EPSON MX80	519

DISK DRIVES

Model III 1-Drive	712
PERCOM TFD 100	389
TEAC 40 Track	319

MODEL I 32K E.I., 1st. Drive, Line Printer IV, 10 Diskettes — Pure R.S. $1875; combined warranty 1699

MODEL II 64K w/2 Disks, Daisy Wheel II, Scripsit — Pure R.S. 6350

MODEL III 48K, 1st. Drive, Line printer IV, Scripsit Word Processor, 10 Diskettes — Pure R.S. 2800; combined warranty 2699

COLOR COMPUTER 16K CC w/extended basic cassette recorder, joy sticks, dust cover — Pure R.S. 569

POCKET COMPUTER w/interface — Pure R.S. 239

ATARI
48K Atari 800, Atari 825 Printer, Atari 810 Disk Drive — 2125

APPLE
48K Apple II & Apple II Disk w/controller, D.O.S. 3.3, Printer Card and Cable, Epson MX80 Printer — 2510

PLUS real back-up warrantees —

Pure Radio Shack equipment warranteed at any Radio Shack store or dealer. Factory warrantees on Apple and Atari equipment. Other equipment carries manufacturer's warranty or Computer Plus 180 day extended warranty. Combined warrantees carry Computer Plus 180 day warranty or original manufacturer's warranty.

DEALER INQUIRIES ARE INVITED (617) 486-3193

Prices subject to change without notice.
Not responsible for typographical errors.

TRS-80 is a registered trademark of Tandy Corp.

call TOLL FREE
1-800-343-8124

computer plus ✓130

Write for your free catalog ...

Dept. D
245A Great Road
Littleton, MA 01460

"As this is. . . outside the ability of a microcomputer, you must decide. . . how trustworthy the Beta is."

```
1710 V$=INKEY$:IF V$="S" GOTO 1740
1720 A=A+1:N=N+1
1730 GOTO 1680
1740 PRINT
1750 CLOSE 1
1760 RETURN
1770 END
```

Chenge Creates Profit—Meybe

Over any period of time the market will rise and fall. We would like to buy when the market is low and sell when the market is high. To do this, we must identity major changes in market direction before they occur or shortly after they begin. The following programs address the subject of major market trends.

Program Listing 4, MOVING/AVE, attempts to indicate major changes in the direction of the market or a stock price by comparing the stock's current price to an average of its past performance. The average of past performance is calculated by finding the average of the first "N" days (normally 200 days are used); then one element is dropped from the front of the series, another element is added to the end of the series and another average is calculated:

First Ave.	Second Ave.	Third Ave.	
1	2	3	This process continues until the
2	3	4	last element of
3	4	5	data stored on
.	.	.	your disk is included in the
.	.	.	average.
200	201	202	
SUM	SUM	SUM:	Divide sum by 200 to obtain the average for each series.

By calculating the moving average, the investor attempts to smooth out the minor changes in a stock's past performance, creating a line that indicates its major trend. By comparing the current stock price to this smooth trend line, the investor can detect any deviations. These deviations are interpreted as buy and sell signals.

Buy Signals:

- If the 200 day average line flattens out or advances following a decline and the daily price of the stock penetrates that average line on the upside.
- If the stock price is above the 200 day line and declines toward it, but fails to go through it, turning up instead.

Sell Signals:

- If the average line flattens out or declines following a rise, and the daily stock price penetrates that line on the downside.
- If the stock price is below the average line and rises toward it, but fails to go through and instead turns down again.[1]

I have found this a very good indicator of major price trend. Of the three programs I have done on this subject, the moving average program has proved much more reliable than the other two.

MOVING/AVE—Sample Execution:

The user has the option of specifying the number of days plotted on the line printer. The computer will instruct you to enter number of days to be plotted to printer. If you wish to see a graph of the last 10 days, enter 10 and they will be plotted. This is primarily a time and paper saving device, elminating duplication when the program is run on a regular basis.

Step 1: ENTER NAME OF DATA FILE? LIONEL/P
Step 2: Computer begins calculating Moving Average (line 130 of program).
Step 3: Graph is plotted on line printer.
P indicates daily stock price on graph.
. indicates Moving Average line.

The other two programs function much like the moving average. One point to keep in mind before going on—these programs attempt to measure market trend and not stock price trend. For this reason, the market index (DJI, Standard & Poor's, etc.)

Slope (Least Squares Line Of Best Fit)

$$= \left[\left(\sum_{i=1}^{N} X \cdot Y \right) - N \cdot \overline{X} \cdot \overline{Y} \right] \Big/ \sum_{i=1}^{N} (Y^2) - (N \cdot \overline{Y}^2)$$

Y Intercept (Least Squares Line Of Best Fit)

$$= \overline{X} - (\text{Slope} \cdot \overline{Y})$$

Correlation Coefficient

$$= \frac{\frac{1}{N-1}\Sigma(X-\overline{X})(Y-\overline{Y})}{\sqrt{\frac{1}{N-1}\Sigma(X-\overline{X})^2}\sqrt{\frac{1}{N-1}\Sigma(Y-\overline{Y})^2}}$$

Terms:
X = Individual Stock Prices
Y = Individual Index Values
$\overline{X}$ = Mean Stock Price
$\overline{Y}$ = Mean Index Value
N = Number of Individual Items

Table 2. FORECAST/MKT Formulas

✓See List of Advertisers on page 306

Program Listing 4. MOVING/AVE indicates major changes in the direction of the market or a stock price, by comparing current stock price to an average of its past performance.

```
10 CLS
20 DIM P(1500),MA(1500)
30 A=1
40 PRINT"PLACE DISK WITH DATA FILE IN DRIVE"
50 INPUT"ENTER NAME OF DATA FILE";A$
60 OPEN"I",1,A$
70 INPUT#1,P(A)
80 IF P(A)=0 GOTO 100
90 A=A+1:GOTO 70
100 CLOSE:CLS
110 AV=0:K=1
120 Q=1
130 A=A-1:N=INT(A/2):IF N>200:N9=N:N=200
140 FOR B=N TO A
150 FOR C=Q TO (Q+N)-1
160 PRINT @ 1,C:PRINT @ 10,B:PRINT @ 20,A
170 AV=P(C)+AV
180 NEXT C
190 AV=AV/N:MA(K)=AV:K=K+1
200 Q=Q+1
210 NEXT B
220 L=5000:H=0
230 L1=5000:H1=0
240 FOR B=1 TO A
250 IF P(B)<L:L=P(B)
260 IF P(B)>H:H=P(B)
270 NEXT
280 FOR B=1 TO K
290 IF MA(B)<L1:L1=MA(B)
300 IF MA(B)>H1:H1=MA(B)
310 NEXT
320 U=L:U1=H
330 R=H-L:R1=H1-L1
340 R=62/R:R1=62/R1
350 LPRINT CHR$(15)A$
360 LPRINT
370 LPRINT INT(N)" DAY MOVING AVERAGE"
380 LPRINTU;:LPRINTTAB(30);" <---------->
                        ";U1
390 LPRINT"-------------------------------------------------
     ---------------------------------"
400 PRINT "ENTER NUMBER OF DAYS TO BE PLOTTED TO PRINTE
     R"
410 PRINT "MUST BE LESS THAN ";INT(A/2)
420 INPUT Z
430 IF Z>= INT(A/2) GOTO 400
440 CLS:IF N>=200 THEN N=N9
450 FOR X=N-Z TO N
460 LPRINT":";
470 T1=R*(P(X)-L):T1=INT(T1)
480 LPRINT TAB(T1)"P"
490 NEXT
500 T=1
510 FOR X=A-Z TO A
520 LPRINT":";
530 T1=R*(P(X)-L):T1=INT(T1)
540 T2=R*(MA(T)-L):T2=INT(T2)
550 T=T+1
560 IF T1=T2 GOTO 620
570 IF T1<T2 GOTO 600
580 LPRINT TAB(T2)".";TAB(T1)"P"
590 GOTO 630
600 LPRINT TAB(T1)"P";TAB(T2)"."
610 GOTO 630
620 LPRINT TAB(T1)"S"
630 NEXT X
640 END
```

Program Listing 5. TRADING/VOL measures the strength or weakness of a given price movement.

```
10 CLS
20 DIM DA(1000),DE(1000)
30 PRINT"PLACE DISK CONTAINING DOW/AVE FILE IN DRIVE"
40 INPUT"PRESS ENTER TO READ DATA FROM DISK.";A$
50 OPEN"I",1,"DOW/AVE"
60 A=1
70 INPUT#1,DA(A)
80 IF DA(A)=0 GOTO 100
```

Program continues

MODEL II

26-4002
64K 1 Drive
$3440.00

26-4160 1 Drive Exp........$1034.00
26-4161 2 Drive Exp.........1574.00
26-4162 3 Drive Exp.........2114.00
26-4530 Scripsit II............265.00
26-4512 Profile II............162.00
26-4511 Visicalc II...........265.00
26-4501 Gen Ledger.........180.00
26-4506 Mail List.............72.00

$ DISCOUNT $

TRS-80®

DEALER A301

COMPUTER SPECIALISTS

CALL US. . .
SAVE MONEY

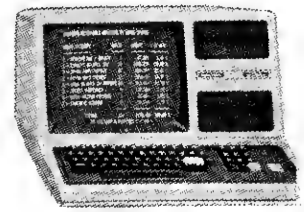
We carry the full line of TRS-80 Computers. All other R.S. software, furniture, and accessories at discount from catalog price. We stock most items to assure you fast delivery and save you money.

26-1140 Expansion Interface.....................$249.00
26-1141 16K Exp. Interface.........................359.00
26-1142 32K Exp. INterface.......................469.00
26-1145 RS232C Board.............................84.00
26-1160/1 Mini Disk Drive.........................419.00
26-1563 Scripsit-Disk..............................79.00
26-1566 Visicalc.....................................83.00

* * * * * ***PRINTERS*** * * * * *

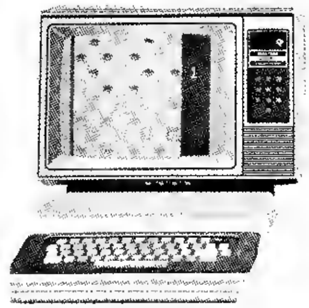
26-1155 Quick Printer...........................187.00
26-1167 9½ Dot Matrix Printer....................360.00
26-1166 Line Printer VI..........................1080.00
26-1158 Daisy Wheel II...........................1799.00
26-1165 Line Printer V...........................1710.00

MODEL III



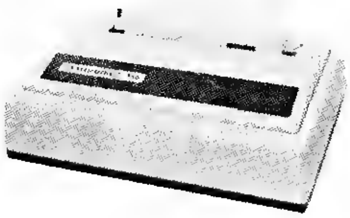
26-1061 4K I.............$629.00
26-1062 16K III...........865.00
26-1063 32K III
W/2 Drives, RS232.......2225.00

COLOR



26-3001 4K..............$353.00
26-3002 16K Ext. Basic.....533.00
26-3008 Joysticks..........22.50
26-3010 Color Video......353.00
26-1206 Recorder..........54.00

PRINTERS

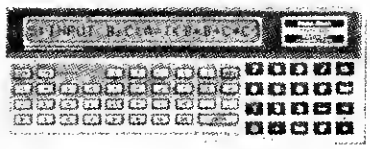


CENTRONICS

BEST PRICES

Fast 100 CPS Centronics
730-1A Printer.............$577.00
Text Quality Centronics
737-1 Printer...............$737.00

Pocket Computer



26-3501 1.9K P.C............$221.00
26-3503 Cassette IF...........45.00
14-812 Recorder..............72.00

commodore

AUTHORIZED DEALER
CALL FOR PRICES

ALL POCKET AND COLOR
COMPUTER SOFTWARE
SOLD AT DISCOUNT

WRITE US FOR A
FREE CATALOG

1-800-841-0860 Toll Free Order Entry

MICRO MANAGEMENT SYSTEMS, INC.

72

No Taxes on Out Of State Shipments

Immediate Shipment From Stock on Most Items

DOWNTOWN PLAZA SHOPPING CENTER
115 C. SECOND AVE. S.W.
CAIRO, GEORGIA 31728
(912) 377-7120 Ga. Phone No. & Export

TRS-80 is a registered trademark of the Tandy Corp.

R.S. 90 Day Limited Warranty
F-48 Form Provided

Largest Inventory
In The S.E. U.S.A.

"As a general rule, the greater the risk, the greater the potential profit."

```
90 A=A+1:GOTO 70
100 CLOSE
110 CLS
120 PRINT"PLACE DISK CONTAINING VOLUME FILE IN DRIVE"
130 INPUT"PRESS ENTER TO READ DATA FROM DISK.";A$
140 B=1
150 OPEN"I",2,"VOLUME"
160 INPUT#2,DE(B)
170 IF DE(B)=0 GOTO 190
180 B=B+1:GOTO 160
190 CLOSE
200 CLS
210 IF A<B THEN L=A-1
220 IF B<A THEN L=B-1
230 IF B=A THEN L=A:GOTO 280
240 FOR D=1 TO L-1
250 DE(D)=DE(D+(B-L))
260 DA(D)=DA(D+(A-L))
270 NEXT
280 L3=500000:H=0:L1=2000:H1=1
290 FOR D=1 TO L-1
300 IF DE(D)>H THEN H=DE(D)
310 IF DE(D)<L3 THEN L3=DE(D)
320 IF DA(D)<L1 THEN L1=DA(D)
330 IF DA(D)>H1 THEN H1=DA(D)
340 NEXT
350 R=H-L3:R2=30/R
360 R1=H1-L1:R3=30/R1
370 LPRINT"VOLUME OF TRADING =  #"
380 LPRINT"DOW AVERAGE =  +"
390 LPRINT"-------------------------------"
392 PRINT"ENTER NUMBER OF DAYS TO BE PLOTED ON PRINTER"
393 PRINT"MUST BE LESS THAN ";L
395 INPUT Z
397 CLS
400 FOR D=L-Z TO L
410 LPRINT":";
420 T1=R3*(DA(D)-L1):T1=INT(T1)
430 T2=R2*(DE(D)-L3):T2=INT(T2)
440 IF T1=T2 GOTO 500
450 IF T1<T2 GOTO 480
460 LPRINT TAB(T2)"#";TAB(T1)"+"
470 GOTO 510
480 LPRINT TAB(T1)"+";TAB(T2)"#"
490 GOTO 510
500 LPRINT TAB(T1)"*"
510 NEXT
520 END
```

Program Listing 6. BREADTH/MKT detects major turning points of the market before or shortly after they occur.

```
10 CLS
20 DIM AD(1000),DE(1000),DA(1000)
30 PRINT"PLACE DISK CONTAINING ADVANCES DATA IN DRIVE"
40 PRINT"FILE MUST BE NAMED 'ADVANCES'."
50 INPUT"PRESS ENTER TO READ DATA";A$
60 OPEN"I",1,"ADVANCES"
70 A=1
80 INPUT#1,AD(A)
90 IF AD(A)=0 GOTO 110
100 A=A+1:GOTO 80
110 CLS
120 CLOSE
130 PRINT"PLACE DISK CONTAINING DECLINES DATA IN DRIVE"

140 PRINT"FILE MUST BE NAMED 'DECLINES'."
150 INPUT"PRESS ENTER TO CONTINUE";A$
160 OPEN"I",2,"DECLINES"
170 B=1
180 INPUT#2,DE(B)
190 IF DE(B)=0 GOTO 210
200 B=B+1:GOTO 180
210 CLOSE
220 CLS
230 PRINT"PLACE DISK CONTAINING DOW AVERAGE DATA IN DRI
 VE"
240 PRINT"FILE MUST BE NAMED 'DOW/AVE'."
250 INPUT"PRESS ENTER TO CONTINUE.";A$
260 OPEN"I",3,"DOW/AVE"
270 C=1
280 INPUT#3,DA(C)
290 IF DA(C)=0 GOTO 310
300 C=C+1:GOTO 280
```

Program continues

Table 3. STOCK/ANA Formulas

Average Annual Rate of Change In Earnings Per Share:

$$\text{Log } A = \frac{\Sigma \text{ Log } Y}{N}$$

$$\text{Log } B = \frac{\Sigma X \cdot \text{Log } Y}{\Sigma X^2}$$

Log Y Est. = Log A + Log B $\cdot$ X
Y Est. = Anti Log of Log Y Est. = Anti Y

Percent Change = Anti Y_{i+1}/Anti Y_i

Average Annual Change = [(Σ Percent Change)/N] − 100

NOTE: Log as used above refers to common logarithms. TRS-80 log function is natural logarithms. Therefore, program converts natural log to common log and then computes anti log of this value.

Y = Earnings Per Share
X = Time
N = Number of EPS values

Beta:

$$(Xi - \bar{X})(Yi - \bar{Y})/(Xi - \bar{X})^2$$

Xi = Investment Performance Relative (IPR) For Index

Yi = Investment Performance Relative (IPR) For Stock

$\bar{Y}$ = Mean IPR For Stock

$\bar{X}$ = Mean IPR For Index

Investment Performance Relative

$$\sum_{i=1}^{n-1} (X_{i+1} + \text{Dividend})/X_i$$

Alpha:

$$\bar{Y} - \frac{(X_i - \bar{X})(Y_i - \bar{Y})}{(X_i - \bar{X})^2} \cdot \bar{X}$$

Coefficient of Determination:

$$\Sigma(\text{Yest} - \bar{Y})^2 / \Sigma(Y - \bar{Y})^2$$

Y = Actual EPS

Yest = See Change In EPS Formula

$\bar{Y}$ = Mean EPS

you use can greatly affect their accuracy.

Program Listing 5, Trading Volume, measures the strength or weakness of a given market movement. Market volume is compared to the Dow Jones Industrial Average (or other index of your choice). This comparison indicates the amount of support behind any market movement.

Index and volume normally move in the same direction. The investor must watch for

"Index and volume normally move in the same direction."

```
310 CLOSE
320 CLS
330 A=A-1:B=B-1:C=C-1
340 PRINT"ALL DATA ENTERED."
350 REM * CREATE DATA FILES OF EQUAL LENGTH."
360 IF A<B OR A<C THEN L=A
370 IF B<A AND B<C THEN L=B
380 IF C<A AND C<B THEN L=C
390 FOR D=1 TO L
400 AD(D)=AD(D+(A-L))
410 DE(D)=DE(D+(B-L))
420 DA(D)=DA(D+(C-L))
430 NEXT
440 FOR D=1 TO L
450 AD(D)=AD(D)-DE(D)
460 NEXT
470 FOR D=1 TO L
480 M=AD(D)+M
490 DE(D)=M
500 NEXT
510 L3=5000:H=-5000
520 L1=2000:H1=1
530 PRINT"HOW MANY DAYS DO YOU WANT PLOTED TO PRINTER"
540 PRINT"MUST BE LESS THAN ";L
550 INPUT Z
560 CLS
570 FOR D= 1 TO L
580 IF DE(D)>H THEN H=DE(D)
590 IF DE(D)<L3 THEN L3=DE(D)
600 IF DA(D)<L1 THEN L1=DA(D)
610 IF DA(D)>H1 THEN H1=DA(D)
620 NEXT
630 R=H-L3:R2=62/R
640 R1=H1-L1:R3=62/R1
650 LPRINT"*************************************************
    ***************************"
660 FOR D= (L-Z) TO L
670 LPRINT":";
680 T1=R3*(DA(D)-L1):T1=INT(T1)
690 T2=R2*(DE(D)-L3):T2=INT(T2)
700 IF T1=T2 GOTO 760
710 IF T1<T2 GOTO 740
720 LPRINT TAB(T2)"B";TAB(T1)"I"
730 GOTO 770
740 LPRINT TAB(T1)"I";TAB(T2)"B"
750 GOTO 770
760 LPRINT TAB(T1)"S"
770 NEXT
```

an index line that is rising while the volume line is falling. Such a condition could indicate that a market rally lacks sufficient support to sustain a prolonged upward trend.

Step 1: PLACE DISK CONTAINING DOW/AVE FILE IN DRIVE PRESS ENTER TO READ DATA FROM DISK ? (enter)
Step 2: PLACE DISK CONTAINING VOLUME FILE IN DRIVE PRESS ENTER TO READ DATA FROM DISK ? (enter)
Step 3: Computer compares index to volume. Volume and index are plotted on line printer.
/ indicates Dow Average on graph.
#Indicates volume of trading.

TRADING/VOL: Sample Program Execution

Program Listing 6, Market Breadth, detects major turning points of the market before or shortly after they occur. Here again, breadth, as measured by advances and declines, and the DJI, typically move together.

When the breadth line declines to new lows while the index is climbing to new highs, the investor must be cautious. This condition could indicate the market is being carried by the large companies that make up the index being used, which in turn could indicate a peak in the index and the ap-

Continue to page 287

Current Ratio:	current assets/ current liabilities
Debt to Equity:	long term debt/ long term debt + owners equity
Net Working Capital:	current assets - current liabilities
Operating Ratio:	total costs and expenses/ net sales
Net Profit to Net Sales:	net income/ net sales
Profits Worth Ratio:	net income/ owners equity
Return on Investment in Assets:	net income/ total assets
Quick Asset Ratio:	cash + marketable securities + accounts and notes receivable/ current liabilities
Inventory Turnover:	cost of goods sold/ (inventory last period + inventory this period / 2)

Table 4. Financial Statement Analysis Formulas

"Interpret the results of this program with extreme caution."

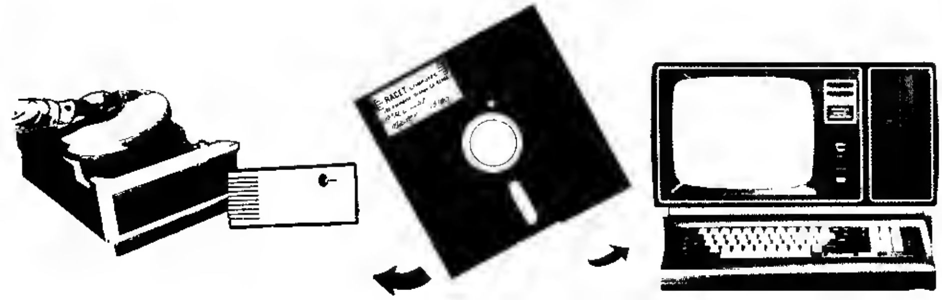
Program Listing 7. FIN/ANA gives the investor an overview of a company's underlying financial strength.

```
10 CLS
20 PRINT "THIS SECTION USES DATA FROM THE BALANCE SHEET
   "
30 PRINT "AND INCOME STATEMENT TO COMPUTE SEVERAL RATIO
   S"
40 PRINT "PERTAINING TO THE FINANCIAL STRENGTH OF THE "

50 PRINT "COMPANY.  INPUT DATA CAN BE TAKEN DIRECTLY FR
   OM"
60 PRINT "THE COMPANYS FINANCIAL REPORT."
70 INPUT "PRESS ENTER TO CONTINUE";A$
80 CLS:INPUT"ENTER CURRENT ASSETS";CA
90 CLS:INPUT"ENTER CURRENT LIABILITIES";CL
100 CLS:INPUT"ENTER LONG TERM DEBT";LD
110 CLS:INPUT"ENTER OWNERS EQUITY";OE
120 CLS:INPUT"ENTER TOTAL COSTS AND EXPENSES";TE
130 CLS:INPUT"ENTER NET SALES";NS
140 CLS:INPUT"ENTER NET INCOME";NI
150 CLS:INPUT"ENTER TOTAL ASSETS";TA
160 CLS:INPUT"ENTER CASH";C
170 CLS:INPUT"ENTER MARKETABLE SECURITIES";MS
180 CLS:INPUT"ENTER ACCOUNTS AND NOTES RECEIVABLE";AR
190 CLS:INPUT"ENTER COST OF GOODS SOLD";CG
200 CLS:INPUT"ENTER PREVIOUS YEAR INVENTORY";I1
210 CLS:INPUT"ENTER CURRENT INVENTORY";I2
220 CLS
230 REM * THE FOLLOWING SECTION COMPUTES RATIOS *
240 WC=CA-CL
250 CR=CA/CL
260 DE=LD/(LD+OE)
270 OP=TE/NS
280 PW=NI/OE
290 PS=NI/NS
300 RI=NI/TA
310 QA=(C+MS+AR)/CL
320 TN=CG/((I1+I2)/2)
330 PRINT TAB(10)"F I N A N C I A L   R A T I O S"
340 PRINT "*********************************************
    ******"
350 PRINT"NET WORKING CAPITAL";TAB(46);WC
360 PRINT"CURRENT RATIO";TAB(46);CR
370 PRINT"DEBT TO EQUITY RATIO";TAB(45);DE
380 PRINT"OPERATING RATIO";TAB(46);OP
390 PRINT"PROFITS-WORTH RATIO";TAB(46);PW
400 PRINT"NET PROFIT TO NET SALES";TAB(46);PS
410 PRINT"RETURN ON INVESTMENT IN ASSETS RATIO";TAB(46)
    ;RI
420 PRINT"QUICK ASSET RATIO";TAB(46);QA
430 PRINT"INVENTORY TURNOVER RATIO";TAB(46);TN
440 PRINT"*********************************************
    *****"
450 INPUT "PRESS ENTER FOR EXPLANATION OF RATIOS";A$
460 CLS
470 PRINT "WHEN INTERPERTING FINANCIAL RATIOS, THERE AR
    E TWO"
480 PRINT "COURSES OF ACTION.  THE FIRST IS TO COMPAIR
    YOUR"
490 PRINT "CALCULATED RATIOS TO THE AVERAGE RATIOS OF T
    HE SAME"
500 PRINT "INDUSTRY.  ONE SOURCE OF INDUSTRY RATIOS IS
    DUN AND"
510 PRINT "BRADSTREETS  'KEY BUSINESS RATIOS'.  IN THIS
    WAY "
520 PRINT "YOU CAN SEE HOW YOUR COMPANY COMPAIRS TO OTH
    ER "
530 PRINT "COMPANIES IN THE SAME INDUSTRY."
540 PRINT "THE SECOND METHOD IS TO CALCULATE THE RATIOS
     FOR"
550 PRINT "YOUR COMPANY OVER A SEVERAL YEAR PERIOD. THI
    S WAY"
560 PRINT "YOU CAN DETECT ANY TRENDS IN THE DIRECTION"
570 PRINT "YOUR RATIOS ARE MOVING AND ACT ACCORDINGLY."

580 INPUT "PRESS ENTER TO CONTINUE";A$
590 CLS
600 PRINT "NET WORKING CAPITAL"
610 PRINT "A MEASURE OF HOW WELL CURRENT OBLIGATIONS AR
    E"
620 PRINT "COVERED BY CURRENT ASSETS.  AS A GENERAL RUL
    E"
630 PRINT "NET WORKINGG CAPITAL SHOULD BE AT LEAST EQUA
    L TO"
640 PRINT "CURRENT LIABILITIES."
650 INPUT "PRESS ENTER TO CONTINUE";A$
660 CLS
670 PRINT "CURRENT RATIO"
```

Program continues

RACET SORTS — RACET UTILITIES — RACET computes — RACET SORTS — RACET UTILITIES — RACET computes — RACET SORTS — RACET UTILITIES — RACET computes

HARD DISK MULTIPLEXOR
FOR TRS-80* Mod II



NOW YOU CAN HAVE THAT LARGE COMMON DATA BASE!!
- Allows up to 4 Mod II's to connect to a single controller — up to 4 hard disk drives per controller. Users may access the same file simultaneously (first-come first-served).
- Uses Cameo controller and standard 10-megabyte cartridge (hard) disk drives along with RACET Hard/Soft Disk System (HSD) software. Removable Disk Pack Backup!
- Access times 3 to 8 times faster than floppy. Mixed floppy/hard disk operation supported.
- Compatible with your existing TRSDOS programs — you need only change filenames! All BASIC statements are identical.
- A single file may be as large as one disk. Directory expandable to handle thousands of files.
- Includes special utilities — XCOPY for backup and copies, XPURGE for multiple deletions, DCS directory catalog system, and Hard Disk Superzap. FORMAT utility includes options for specifying sectors/gran, platters/drive, logical disk size, etc.

HARD DISK DRIVE AND CONTROLLER $5995 **RACET HSD Software $400** Call for multiuser pricing. Dealers call for OEM pricing.

★★NEW★★ DISCAT (32K 1-drive Min) **Mod I, III $50.00**
This comprehensive Diskette Cataloguing/Indexing utility allows the user to keep track of thousands of programs in a categorized library. Machine language program works with all TRSDOS and NEWDOS versions. Files include program names and extensions, program length, diskette numbers, front and back, and diskette free space.

★★NEW★★ KFS-80 (1-drive 32K Min — Mod II 64K) **Mod I, III $100.00; Mod II $175.00**
The keyed file system provides keyed and sequential access to multiple files. Provides the programmer with a powerful disk handling facility for development of data base applications. Binary tree index system provides rapid access to file records.

★★NEW★★ MAILLIST (1-drive 32K Min — Mod II 64K) **Mod I, III $75.00; Mod II $150.00**
This ISAM-based maillist minimizes disk access times. Four keys — no separate sorting. Supports 9-digit zip code and 3-digit state code. Up to 30 attributes. Mask and query selection. Record access times under 4 seconds!!

★★NEW★★ LPSPOOL (32K 1-drive Min) **Mod I $75.00**
LPSPOOL — Add multi-tasking to permit concurrent printing while running your application program. The spooler and despooler obtain print jobs from queues maintained by the system as print files are generated. LPSPOOL supports both parallel and serial printers.

BASIC LINK FACILITY 'BLINK' (Mod I Min 32K 1-disk) **Mod I $25.00; Mod II $50.00; Mod III $30.00**
Link from one BASIC program to another saving all variables! The new program can be smaller or larger than the original program in memory. The chained program may either replace the original program, or can be **merged** by statement number. The statement number where the chained program execution is to begin may be specified!

INFINITE BASIC (Mod I & Mod III Tape or Disk) **Mod I $50.00; Mod III $60.00**
Extends Level II BASIC with complete MATRIX functions and 50 more string functions. Includes RACET machine language sorts! Sort 1000 elements in 9 seconds!! Select only functions you want to optimize memory usage.

INFINITE BUSINESS (Requires Infinite BASIC) **Mod I & III $30.00**
Complete printer pagination controls — auto headers, footers, page numbers. Packed decimal arithmetic — 127 digit accuracy +, −, *, /. Binary search of sorted and unsorted arrays. Hash codes.

COMPROC (Mod I & Mod III — Disk only) **Mod I $20.00; Mod III $30.00**
Command Processor. Auto your disk to perform any sequence of instructions that you can give from the keyboard. DIR, FREE, pause, wait for user input, BASIC, No. of FILES and MEM SIZE, RUN program, respond to input statements, BREAK, return to DOS, etc. Includes lowercase driver software, debounce and screenprint!

GSF (Mod I & III Tape or Disk — Specify Memory Size) **Mod I $25.00; Mod II $50.00; Mod III $30.00**
Generalized Subroutine Facilities. The STANDARD against which all other sorts are compared! Machine language — fast and powerful! Multi-key multi-variable and multi-key character string. Zero and move arrays. Mod II includes USR PEEKS and POKES. Includes sample programs.

DSM (Mod I Min 32K 2-drive system. Mod II 64K 1-drive. Mod III Min 32K 1-drive) **Mod I $75.00; Mod II $150.00; Mod III $90.00**
Disk Sort/Merge for RANDOM files. All machine language stand-alone package for sorting speed. Establish sort specification in simple BASIC command File. Execute from DOS. Only operator action to sort is to change diskettes when requested! Handles multiple diskette files! Super fast sort times — improved disk I/O times make this the fastest Disk Sort/Merge available on the TRS.

UTILITY PACKAGE (Mod II 64K) **$150.00**
Important enhancements to the Mod II. The file recovery capabilities alone will pay for the package in even one application! Fully documented in 124 page manual! XHIT, XGAT, XCOPY and SUPERZAP are used to reconstruct or recover data from bad diskettes! XCOPY provides multi-file copies, 'wild-card' mask select, absolute sector mode and other features. SUPERZAP allows examine/change any sector on diskette including track-0, and absolute disk backup/copy with I/O recovery. DCS builds consolidated directories from multiple diskettes into a single display or listing sorted by disk name or file name plus more. Change Disk IO with DISKIO. XCREATE preallocates files and sets 'LOF' to end to speed disk accesses. DEBUGII adds single step, trace, subroutine calling, program looping, dynamic disassembly and more!!

BASIC CROSS REFERENCE UTILITY (Mod II 64K) **$50.00**
SEEK and FIND functions for Variables, Line Numbers, Strings, Keywords. 'All' options available for line numbers and variables. Load from BASIC — Call with 'CTRL'R. Output to screen or printer!

DEVELOPMENT PACKAGE (Mod II 64K) **$125.00**
Includes RACET machine language SUPERZAP, Apparat Disassembler, and Model II interface to the Microsoft 'Editor Assembler Plus' software package including uploading services and patches for Disk I/O. Purchase price includes complete copy of Editor Assembler + and documentation for Mod I. Assemble directly into memory, MACRO facility, save all or portions of source to disk, dynamic debug facility (ZBUG), extended editor commands.

CIRCLE READER REQUEST FOR FREE 24-PAGE CATALOG
*TRS-80 IS A TRADEMARK OF TANDY CORPORTION

CHECK, VISA, M/C, C.O.D., PURCHASE ORDER
TELEPHONE ORDERS ACCEPTED (714) 637-5016

RACET COMPUTES
1330 N. GLASSEL, SUITE 'M', ORANGE, CA 92665

✓41

RACET SORTS — RACET UTILITIES — RACET computes — RACET SORTS — RACET UTILITIES — RACET computes — RACET SORTS RACET SORTS — RACET UTILITIES —

✓See List of Advertisers on page 306

". . . there is always . . . risk . . . never gamble unless you can afford to lose."

```
680 PRINT "A MEASURE OF A COMPANYS ABILITY TO MEET ITS
     DAY"
690 PRINT "TO DAY EXPENSES."
700 INPUT "PRESS ENTER TO CONTINUE";A$
710 CLS
720 PRINT "DEBT TO EQUITY RATIO"
730 PRINT "COMPAIRS BORROWED FUNDS TO OWNERSHIP FUNDS.
     "
740 PRINT "GENERALLY A LOWER DEBT TO EQUITY RATIO IS LE
     SS"
750 PRINT "RISKY."
760 INPUT "PRESS ENTER TO CONTINUE";A$
770 CLS
780 PRINT "OPERATING RATIO"
790 PRINT "THE PROPORTION OF COSTS NEEDED TO PRODUCE EA
     CH DOLLAR"
800 PRINT "OF SALES.  A LOWER RATIO MEANS A LOWER EXPEN
     DITURE"
810 PRINT "OF COST AND EXPENSE DOLLARS TO CREATE EACH D
     OLLAR"
820 PRINT "OF SALES."
830 INPUT "PRESS ENTER TO CONTINUE";A$
840 CLS:PRINT"PROFITS WORTH RATIO"
850 PRINT " A MEASURE OF A COMPANYS PROFITABILITY.  THE
       HIGHER"
860 PRINT "THE RATIO THE GREATER THE AMOUNT OF PROFIT P
     RODUCED"
870 PRINT "BY THE OWNERSHIP INVESTMENT."
880 INPUT "PRESS ENTER TO CONTINUE";A$
890 CLS:PRINT "NET PROFIT TO NET SALES"
900 PRINT "THE AMOUNT REMAINING OF EACH SALES DOLLAR AF
     TER ALL"
910 PRINT "COSTS, EXPENSES AND TAXES HAVE BEEN PAID.  T
     HE HIGHER"
920 PRINT "THE RATIO, THE GREATER THE PROFIT."
930 INPUT "PRESS ENTER TO CONTINUE";A$
940 CLS:PRINT "RETURN ON INVESTMENT"
950 PRINT "AMOUNT RETURNED FOR EACH DOLLAR INVESTED IN
     THE"
960 PRINT "COMPANYS RESOURCES.  THE HIGHER THE RATIO, T
     HE "
970 PRINT "HIGHER THE RETURN FROM INVESTMENT IN ASSETS.
     "
980 INPUT "PRESS ENTER TO CONTINUE";A$
990 CLS:PRINT "QUICK ASSET RATIO"
1000 PRINT "A MEASURE OF A COMPANYS ABILITY TO PAY ITS
      DEBTS."
1010 PRINT "NORMALLY A RATIO OF 1.0 OR HIGHER IS SUFFIC
      IENT."
1020 INPUT "PRESS ENTER TO CONTINUE";A$
1030 CLS:PRINT "INVENTORY TURNOVER"
1040 PRINT "EVALUATES THE EFFECTIVENESS OF INVENTORY MA
      NAGEMENT."
1050 PRINT "SHOWS OVER OR UNDER INVESTMENT IN INVENTORY
       AND HOW"
1060 PRINT "SALEABLE THE GOODS ARE.  MEASURES THE NUMBE
      R OF TIMES"
1070 PRINT "A COMPANYS INVENTORY IS REPLACED DURING THE
       YEAR."
1080 INPUT "PRESS ENTER TO CONTINUE";A$
1090 CLS:GOTO 330
1100 END
```

FIN/ANA Sample Execution

Step 1: THIS PROGRAM USES DATA FROM THE BALANCE SHEET AND INCOME STATEMENT TO COMPUTE SEVERAL RATIOS PERTAINING TO THE FINANCIAL STRENGTH OF THE COMPANY. INPUT DATA CAN BE TAKEN DIRECTLY FROM THE COMPANY'S FINANCIAL REPORT.
PRESS ENTER TO CONTINUE.

Step 2:	ENTER CURRENT ASSETS?	278.6
	ENTER CURRENT LIABILITIES	87.71
	ENTER LONG TERM DEBT?	95.60
	ENTER OWNERS EQUITY?	194.8
	ENTER TOTAL COSTS AND EXPENSES?	70.70
	ENTER NET SALES?	579.1
	ENTER NET INCOME?	20.45
	ENTER TOTAL ASSETS?	392.6
	ENTER CASH?	37.13
	ENTER MARKETABLE SECURITIES?	19.83
	ENTER ACCOUNTS AND NOTES RECEIVABLE?	19.83

Sample continues

```
        ENTER COSTS OF GOODS SOLD?            528.2
        ENTER PREVIOUS YEAR INVENTORY?        176.4
        ENTER CURRENT INVENTORY?              208.8
```

Step 3: Computer calculates financial ratios.

```
                FINANCIAL RATIOS

NET WORKING CAPITAL                          190.9
CURRENT RATIO                                3.176
DEBT TO EQUITY RATIO                         .3292
OPERATING RATIO                              .1220
PROFITS/WORTH RATIO                          .1049
NET PROFIT TO NET SALES                      .0353
RETURN ON INVESTMENT IN ASSETS RATIO         .0520
QUICK ASSET RATIO                            .9521
INVENTORY TURNOVER RATIO                     2.742
PRESS ENTER TO CONTINUE
```

Step 4: Computer explanation of financial ratios.

WHEN INTERPRETING FINANCIAL RATIOS, THERE ARE TWO COURSES OF ACTION. THE FIRST IS TO COMPARE YOUR CALCULATED RATIOS TO THE AVERAGE RATIOS OF THE SAME INDUSTRY. ONE SOURCE OF INDUSTRY RATIOS IS DUN AND BRADSTREETS 'KEY BUSINESS RATIOS.' IN THIS WAY YOU CAN SEE HOW YOUR COMPANY COMPARES TO OTHER COMPANIES IN THE SAME INDUSTRY.

THE SECOND METHOD IS TO CALCULATE THE RATIOS FOR YOUR COMPANY OVER A SEVERAL YEAR PERIOD. THIS WAY YOU CAN DETECT ANY TRENDS IN THE DIRECTION YOUR RATIOS ARE MOVING AND ACT ACCORDINGLY.

PRESS ENTER TO CONTINUE

NET WORKING CAPITAL:

A MEASURE OF HOW WELL CURRENT OBLIGATIONS ARE COVERED BY CURRENT ASSETS. AS A GENERAL RULE NET WORKING CAPITAL SHOULD BE AT LEAST EQUAL TO CURRENT LIABILITIES.

PRESS ENTER TO CONTINUE

CURRENT RATIO:

A MEASURE OF A COMPANY'S ABILITY TO MEET ITS DAY TO DAY EXPENSES.

PRESS ENTER TO CONTINUE

DEBT TO EQUITY RATIO:

COMPARE BORROWED FUNDS TO OWNERSHIP FUNDS. GENERALLY A LOWER DEBT TO EQUITY RATIO IS LESS RISKY.

PRESS ENTER TO CONTINUE

OPERATING RATIO:

THE PROPORTION OF COSTS NEEDED TO PRODUCE EACH DOLLAR OF SALES. A LOWER RATIO MEANS A LOWER EXPENDITURE OF COST AND EXPENSE DOLLARS TO CREATE EACH DOLLAR OF SALES.

PRESS ENTER TO CONTINUE

PROFITS WORTH RATIO:

A MEASURE OF A COMPANY'S PROFITABILITY. THE HIGHER THE RATIO THE GREATER THE AMOUNT OF PROFIT PRODUCED BY THE OWNERSHIP INVESTMENT.

PRESS ENTER TO CONTINUE

NET PROFIT TO NET SALES:

THE AMOUNT REMAINING OF EACH SALES DOLLAR AFTER ALL COSTS, EXPENSES AND TAXES HAVE BEEN PAID. THE HIGHER THE RATIO, THE GREATER THE PROFIT.

PRESS ENTER TO CONTINUE

RETURN ON INVESTMENT:

AMOUNT RETURNED FOR EACH DOLLAR INVESTED IN THE COMPANY'S RESOURCES. THE HIGHER THE RATIO, THE HIGHER THE RETURN FROM INVESTMENT IN ASSETS.

PRESS ENTER TO CONTINUE

QUICK ASSET RATIO:

A MEASURE OF A COMPANY'S ABILITY TO PAY ITS DEBTS. NORMALLY A RATIO OF 1.0 OR HIGHER IS SUFFICIENT.

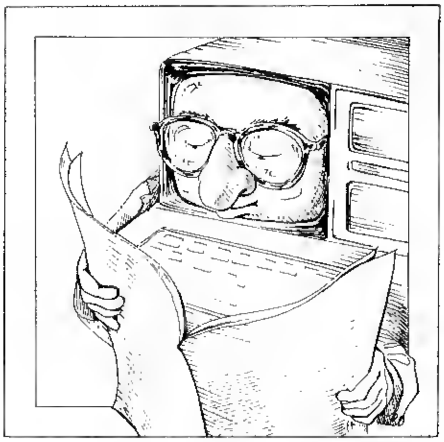
PRESS ENTER TO CONTINUE

INVENTORY TURNOVER:

EVALUATES THE EFFECTIVENESS OF INVENTORY MANAGEMENT. SHOWS OVER OR UNDER INVESTMENT IN INVENTORY AND HOW SALABLE THE GOODS ARE. MEASURES THE NUMBER OF TIMES A COMPANY'S INVENTORY IS REPLACED DURING THE YEAR.

PRESS ENTER TO CONTINUE

Step 5: Computer displays financial ratios on screen again.



Step 1: PLACE DISK CONTAINING ADVANCES DATA IN DRIVE FILE MUST BE NAMED 'ADVANCES'.
PRESS ENTER TO READ DATA? (enter)

Step 2: PLACE DISK CONTAINING DECLINES DATA IN DRIVE FILE MUST BE NAMED 'DECLINES'.
PRESS ENTER TO CONTINUE? (enter)

Step 3: PLACE DISK CONTAINING DOW AVERAGE DATA IN DRIVE FILE MUST BE NAMED 'DOW/AVE'.
PRESS ENTER TO CONTINUE? (enter)

Step 4: Computer calculates breadth of market by subtracting declines from advances for each day and adding the difference. Advances and declines are the daily advance and decline figures which can be found in most major newspapers.

Step 5: Computer compares breadth to index, plotting output to line printer.
B indicates breadth on graph.
I indicates index.

BREADTH/MKT—Sample Execution

Invest in a Company

We have thus far limited our analysis to stock price and market value, which tells us nothing of the financial strength of the company in which we are about to invest. When dealing in the stock market, always remember you are buying part of a company. Stock price alone doesn't tell you what a company is worth, it tells you what someone else is willing to pay for part of that company. The final program gives us a look at what we are buying.

Program Listing 7, Financial Analysis, gives the investor an overview of a company's underlying financial strength by calculating the common financial ratios that measure its economic activity. Each ratio is explained during program execution.

The data used by this program must be entered through the keyboard. The data can be taken directly from a company's annual report (income statement and balance sheet) or one of several reference books.

Conclusion

Remember, these programs do not make a decision for you; they merely supply you with information that will help you make a more informed decision. The decision is left to you. After all, it is your money. ■

[1]*Guide to Intelligent Investing*, Jerome B. Cohen, Edward D. Zinberg, Arthur Zeikel. Dow-Jones Irwin, Publishers.

From page 283

proach of a down trend in stock prices. In other words, major investors are avoiding the riskier (high Beta) small companies and sticking with the stabler large companies, in anticipation of a market downturn.

Using the Rochester Data I/O Pak in the real world.

Electro-Mechanical Hard Copy

Sherman Levine
84 Greenwood Lane
White Plains, NY 10607

Despite the fact that dot matrix printers are considerably faster and less expensive than solid face printers, most business correspondence is written using solid face type, even when high-speed dot matrix printers are readily available. Those of us who use word processors for preparing text and correspondence need a method for preparing typewriter-quality text from computer output.

Since I am an avid writer, I needed a relatively inexpensive system which was simple to use and program (I had never done any machine language programming), capable of storing text rapidly and printing typewriter-quality output. I already had access to several high quality electric typewriters, and did not need rapid printing so that the purchase of a new solid face printer seemed excessive.

After much consideration, I purchased a system consisting of the TRS-80 (Level II, 16K), the Exatron Stringy Floppy, and the Rochester Data I/O Pak typewriter driver, (now called the Dynatyper), along with the Electric Pencil as my word processor. This system has a number of advantages over the other combinations I considered:

- Cost: It does not require an expansion interface, since disk drives are not used, and the I/O Pak is considerably less expensive than new computer driven Selectric or Daisy Wheel printers.
- Ease of programming: The Exatron Stringy Floppy has been described in detail before (see *80 Microcomputing* May, 1980). It fits my needs because it saves and loads both BASIC and machine language programs easily and quickly, without requiring a great deal of programming skill. In addition, Exatron supplies Electric Pencil for the Stringy Floppy for a few dollars more than the standard tape version.

Hardware

The I/O Pak ($469) consists of an array of solenoid coils which fit over and depress the keys of any standard electric typewriter that has a powered carriage return. The solenoids are driven by a six bit code generated by a short software program residing in upper memory (more about this later). When the solenoid coil is actuated, a small metal core attached to a Delrin rod pushes down the selected typewriter key and the character is printed. The unit rests on two mounting sites which are fastened to the keyboard using double-sided tape, so that unit can be removed for standard typewriter use, and thus, is useable on any typewriter.

The solenoid unit connects to the computer through an interface ($80) which decodes the eight bit address bus and triggers a one-shot multivibrator when the correct address and out signals occur simultaneously. This unit uses the lower six bits of the data bus to feed two line drivers, the output of which is arranged in an eight by eight matrix which drives the solenoids through transistor buffers. The use of the one-shot simplifies software timing and ensures that program errors will never lock a solenoid in an energized state. Both the interface and the Stringy Floppy plug into the keyboard expansion connector through an extension cable supplied with the Stringy Floppy.

```
1060 DATA 241,254,13,40,36,0,0,0
1070 DATA 00,254,8,40,38,254,9,40
```

Program Listing 1. Modifications to Rochester Data program to speed typing spaces.

The power supply ($66) provides ±18 VDC for the solenoids and +5 VDC for the TTL integrated circuits in the interface. The circuitry for both the interface and the power supply are included in the I/O PAK documentation for those who wish to build their own.

Software

A BASIC program which creates a relocatable 256 byte machine program residing in upper memory is available from Rochester Data either as a listing (free) or on cassette ($15). The program permits printing BASIC programs, with both uppercase and lowercase for strings, using the LLIST and LPRINT commands. The program has variable timing for the tab, backspace, space, character and carriage return functions.

After you are satisfied with the timing constants and codes, the driver program can be saved using the Stringy Floppy @SAVE command (@SAVEn,32512,256

for 16K machines) and loaded whenever needed without rerunning the entire BASIC program. The commands POKE 16422,0: POKE 16423,127 which define the location of the new printer driver should be entered manually at the start of program loading or included in any BASIC program which uses LPRINT. To improve the speed of the space timing, I suggest replacing four bytes of lines 1060–1070 of the Rochester Data program with zeroes, as shown in Program Listing 1.

The machine language printer driver stores the key code for each ASCII character as an eight bit code in a lookup table. The high-order two bits define the case of the character (00 case independent, for example 'period' 10 uppercase on typewriter 01 lowercase on typewriter). The other six bits are used to select the correct character solenoid, via an eight by eight matrix in the interface. The program examines the case of each character to be printed, and compares it to the state of the shift mechanism. If a shift change is necessary, either the shift lock (to enter uppercase) or shift (to enter lowercase) is momentarily depressed, followed by the character to be printed.

In addition, each character output by the computer initiates a short timing loop, equal to the delay time between typed characters, which prevents return to the main program until the delay is complete. The length of this timing loop depends upon the typewriter used, and whether the character is a shift, carriage return, backspace, etc. The one-shot in the interface pulses for a time which is much shorter than this delay, since the solenoid action is needed only to initiate character printing.

Patches to Electric Pencil

I chose the Electric Pencil program because it was fast, flexible, and short—only 5300 bytes, leaving 11360 bytes (about seven to eight double spaced pages) for text storage. This is enough for letters and short manuscripts. (SCRIPSIT, by comparison, leaves only about 4K bytes for text storage.) For longer manuscripts, I store each section on a separate 20 foot Stringy Floppy tape, and combine material at the time of printing only. Since the Electric Pencil is written in machine language, and is undocumented, figuring out the patch to the printer driver required a bit of work, especially since the Electric Pencil does not have provisions for stopping at the end of each page for paper changes.

LOCATION DECIMAL	LOCATION HEX	COMMAND DECIMAL	COMMAND HEX	EXPLANATION
21139	5293	58	3A	LD A,(nn) loads contents of
21140	5294	232	E8	memory location 37E8 into
21141	5295	55	37	register A.
21142	5296	254	FE	CP(n) compares the contents of
21143	5297	255	FF	register A with FF.
21157	52A5	58	3A	LD A,(nn)—loads contents of
21158	52A6	232	E8	mem location 37E8 (printer
21159	52A7	55	37	port) into register A
21160	52A8	230	E6	AND F0—performs logical AND of
21161	52A9	240	F0	A and F0 (lower 4 bits = 0)
21162	52AA	254	FE	CP(n)—compares the result to
21163	52AB	48	30	30.
21164	52AC	32	20	JRNZ F7—If A reg is not 30,
21165	52AD	247	F7	jumps back to 52A5.
21166	52AE	241	F1	POP AF—takes char from stack
21167	52AF	50	32	LD (nn),A—loads contents of A
21168	52B0	232	E8	reg into memory location
21169	52B1	55	37	37E8 (printer port).
21170	52B2	201	C9	RET—returns to program.

Program Listing 2. References to Memory Location 37E8 in Electric Pencil

How does the Electric Pencil or other programs interact with the printer? It seemed most reasonable to me that the program must output data to the printer and monitor the printer port to see if it's ready to accept the next character. Since the line printer address is 37E8 (hex), references to this location would be of the form:

instruction
low order byte (E8 hex = 232 decimal)
high order byte (37 hex = 55 decimal)

After loading the Electric Pencil program, I ran the following, searching for bytes 232 and 55 in that sequence:

```
FOR J = 17232 TO 21408:IF PEEK(J) = 232
AND PEEK(J + 1) = 55 THEN PRINTJ ELSE
NEXT (ENTER)
```

(17232 and 21408 are the start and end points of the Pencil, according to the ESF Monitor program.) Three sets of references to 37E8 were found at the following locations: 21140 (5294 hex), 21158 (52A6 hex) and 21168 (52B0 hex). Examination of the instructions at these locations using the ESF Monitor program revealed the sequence shown in Program Listing 2. The set of instructions beginning at 52A5 examines the contents of memory location 37E8 (actually the print-

✓See List of Advertisers on page 306

Program Listing 3. Electric Pencil Patches to Printer Driver

1. Load program from wafer. Do not run.

2. Enter the following commands from the keyboard:

   POKE 21139, 62 (ENTER)
   POKE 21140, 48 (ENTER)
   POKE 21141, 0 (ENTER)

   POKE 21157, 62 (ENTER)
   POKE 21158, 48 (ENTER)
   POKE 21159, 0 (ENTER)

   POKE 21167,205 (ENTER) (This is subroutine call)
   POKE 21168,220 (ENTER) (Lower byte of driver address)
   POKE 21169,126 (ENTER) (Upper byte of driver address)

3. Save program on wafer (@SAVEn,17232,4176) without autostart. (@SAVEn, 17232,4176,17232) with autostart.

Program Listing 4.

```
10 REM PRINTER DRIVER FOR PENCIL AND BASIC PROGRAMS
20 REM          S.D. LEVINE.  JULY 198Ø
30 REM FOR PENCIL PROGRAMS CHANGE
40 REM LINE 35Ø TO:   35Ø DATA Ø
60 REM   FOR BASIC USE AS IS (START ADDRESS 7EDC=32476)
70 REM   OR DELETE LINES 37Ø-42Ø AND CHANGE START ADDRESS
80 REM   TO 7F19 = 32537.
85 DIMV(42):CLS:READ BP:AD=BP
86 IF PEEK(16561)+256*PEEK(16562)<BP GOTO 9Ø
87 PRINT "YOU NEED MEMORY SIZE",BP-1,"OR LESS"
88 PRINT "START AGAIN. YOU SHOULD ENTER:"
89 PRINT "SYSTEM (ENTER)":PRINT "/Ø (ENTER)":END
90 POKE 16423,INT(BP/256):POKE 16422,(BP-256*PEEK(16423))
91 PRINT "START ADDRESS =",BP
92 PRINT "TO USE FOR BASIC PROGRAMS POKE THE FOLLOWING AT THE
93 PRINT "START OF PROGRAM LOADING OR AT THE START OF EACH
94 PRINT "PROGRAM REQUIRING PRINTING:
95 PRINT "POKE 16422,";PEEK(16422);":POKE 16423,";PEEK(16423)
100 READ B$:IFB$="XX"THEN14Ø
110 A$=LEFT$(B$,1):IFA$="L"GOSUB26Ø  :V(N)=AD:GOTO1ØØ
120 IF A$="T"THENAD=AD+1
130 AD=AD+1:GOTO1ØØ
140 AD=BP:RESTORE:READ BP:PRINT "FIRST PASS DONE"
150 READ B$: IFB$="XX"THEN 27Ø  ELSE A$=LEFT$(B$,1)
160 IF A$="L" THEN 15Ø  ELSE IF A$="T"THEN 22Ø  ELSE IF A$="R"THEN24Ø
170 GOSUB2ØØ  :A=16*X:A$=RIGHT$(B$,1):GOSUB2ØØ  :A=A+X
180 POKE AD,A:AD=AD+1:GOTO15Ø
200 X=ASC(A$)-48:IFX>9THENX=X-7
210 RETURN
220 GOSUB26Ø  :GOSUB23Ø  :AD=AD+2:GOTO15Ø
230 A=INT(V(N)/256):POKE AD+1,A:POKE AD,V(N)-256*A:RETURN
240 GOSUB26Ø  :A=V(N)-AD-1:IFA<ØTHENA=256+A
250 GOTO18Ø
260 N=ASC(RIGHT$(B$,1))-48:RETURN
270 REM  THIS IS THE LINE THE PROGRAM JUMPS TO WHEN 2ND PASS
280 REM  IS COMPLETE.  DATA LINE BEGINS WITH STARTING ADDRESS
290 REM  IN DECIMAL, THEN CONTINUES WITH INSTRUCTIONS IN HEX
300 REM  AS OUTLINED BY L. SUTER (8Ø MICRO, 4/8Ø).  LAST
301 REM DATA ITEM IS 'XX'.
302 PRINT"TIME DELAY FOR CHAR=";PEEK(V(3Ø)+17);" AT";V(3Ø)+17
303 PRINT"ADDED DELAY FOR RPT=";PEEK(V(3Ø)+12);" AT";V(3Ø)+12
304 PRINT"DELAY FOR CARR RET =";PEEK(V(3Ø)+27);" AT";V(3Ø)+27
305 PRINT"DELAY FOR BACKSPACE=";PEEK(V(3Ø)+36);" AT";V(3Ø)+36
306 PRINT"DELAY FOR SHIFT=    ";PEEK(V(3Ø)-5); " AT";V(3Ø)-5
307 PRINT "MAKE ANY CHANGES NOW USING POKE COMMAND":END
324 REM
325 REM THIS IS STARTING ADDRESS
330 DATA 32476
338 REM
339 REM PUSHES REGISTERS ONTO STACK
340 DATA E5,C5,F5
348 REM
349 REM C REG TO A REG. SET TO Ø (NO OP) FOR PENCIL
350 DATA 79
358 REM
359 REM LOWER 7 BITS OF A REG TO STACK
360 DATA E6,7F
367 REM
368 REM COMPARES A REG TO 2ØHEX (SPACE). IF SAME JUMPS TO
369 REM "X". LOWER 5 BITS OF A REG SET TO ZERO
370 DATA F5,FE,2Ø,28,RX,E6,EØ
371 REM
372 REM PUSH AF.
373 REM IF REMAINING BITS OF A ARE NONZERO (A IS NOT A CONTROL
374 REM CHARACTER), THEN JUMP TO "A". IF A REG IS ZERO (A IS A
375 REM CONTROL CHARACTER) MOVE LINEFEED COUNT "R" TO A,
376 REM INCREMENT A AND COMPARE TO 8. (HAVE 8 CONTROL CHARS FOR
377 REM 4 LINES). iF A=8 JUMP TO NEXT DATA LINE, OTHERWISE
378 REM MOVE NEW LINEFEED COUNT FROM A TO "R" AND JUMP TO "X"
380 DATA 2Ø,RA,3A,TR,3C,FE,Ø8,28,Ø5,32,TR,18,RX
381 REM
382 REM WE GOT HERE BECAUSE CONTROL CHAR WAS OUTPUT AND
383 REM LINEFEED COUNT WAS 8. PUT 1 INTO A REGISTER. MOVE
384 REM CONTENTS OF A TO "F" (FLAG). JUMP TO "X".
385 REM
386 REM LABEL "A". IF WE GET HERE, IT MEANS THAT CHARACTER
387 REM OUTPUT WAS NOT A CONTROL CHARACTER. PUT Ø INTO A
388 REM REGISTER, AND MOVE THIS Ø FROM A REG INTO LINEFEED
389 REM COUNT REGISTER. MOVE CONTENTS OF FLAG REG "F" TO A REG
390 DATA 3E,Ø1,32,TF,18,RX,LA,AF,32,TR,3A,TF
391 REM
392 REM COMPARE CONTENTS OF A REG (WHICH HOLDS FLAG) TO Ø.
393 REM IF A REG=Ø, THEN JUMP TO "X" AND CONTINUE PRINTING.
394 REM OTHERWISE, WAIT UNTIL EITHER SPACE OR BREAK IS PRESSED:
395 REM LABEL "S"
396 REM MOVE CONTENTS OF MEMORY LOCATION 384Ø TO A REG.
397 REM IF A REG NOT EQUAL TO 8Ø (SPACE BAR NOT PRESSED)
398 REM THEN JUMP DOWN TO SEE IF BREAK KEY PRESSED.
399 REM OTHERWISE (SPACE BAR PRESSED) PUT Ø INTO A REG.
400 DATA FE,ØØ,28,RX,LS,3A,4Ø,38,FE,8Ø,2Ø,Ø6,AF
401 REM
402 REM MOVE Ø FROM A REGISTER INTO FLAG LOCATION "F".
403 REM JUMP TO "X" AND CONTINUE WITH PRINTING NEW PAGE
404 REM
405 REM CHECK TO SEE IF BREAK KEY PRESSED BY COMPARING
406 REM A REGISTER TO 4. iF BREAK KEY NOT PRESSED, JUMP BACK
407 REM TO "S". iF BREAK KEY PRESSED, POP A REG AND JUMP TO "U"
408 REM AND RETURN TO PENCIL PROGRAM WITHOUT FURTHER PRINTING
410 DATA 32,TF,18,RX,FE,Ø4,2Ø,RS,F1,18,RU
411 REM
412 REM LABEL "X". POP AF. THIS IS THE LAST LINE TO BE DELETED
413 REM IF YOU WANT TO USE THIS FOR "BASIC" PROGRAMS INSTEAD OF
414 REM THE ELECTRIC PENCIL.
420 DATA LX,F1
421 REM MOVE A REG TO C REG. MOVE Ø TO B REG. MOVE LOCATION OF
422 REM FIRST BYTE OF LOOKUP TABLE (LABEL L) TO HL REG.
424 REM ADD CONTENTS OF BC REG AND HL REG, AND STORE IN HL REG.
425 REM SINCE A REG INITIALLY HELD ASCII CODE, NUMBER NOW IN HL
426 REM REG IS LOCATION OF TYPEWRITER CODE FOR CHARACTER TO BE
427 REM PRINTED. MOVE CONTENTS OF MEMORY LOC WHOSE ADDRESS
428 REM IS IN THE HL REG INTO REG C. PUT CØ INTO REGISTER A.
429 REM PUT C REG AND CO (THAT IS, TOP 2 BITS OF C REG) INTO A.
430 DATA 4F,Ø6,ØØ,21,TL,Ø9,4E,3E,CØ,A1
431 REM
432 REM IF A REG IS ZERO, THEN TYPEWRITER CODE IS CASE-
433 REM INDEPENDENT. DO NOT CHANGE CASE. JUMP TO "N".
434 REM OTHERWISE MOVE A REG TO B REG. MOVE CASE CODE FROM
435 REM "K" TO A. COMPARE TO B REG. IF THEY'RE THE SAME, DO
436 REM NOT CHANGE CASE. JUMP TO "N". IF THEY'RE DIFFER
```

Program continues

```
        ENT,
437 REM MUST CHANGE CASE. MOVE B REG TO A REG, STORE A
        REG
438 REM (NEW CASE CODE) IN "K". "AND" A REG WITH
439 REM 40 (0100 0000).
440 DATA 28,RN,47,3A,TK,B8,28,RN,78,32,TK,E6,40
441 REM
442 REM PUSH BC ONTO STACK. PUT 0E (0000 1110) INTO C R
        EG.
443 REM IF A REG WAS EQUAL TO 40, THEN NEW CHAR IS UPPE
        R CASE
444 REM AND MUST SET SHIFT LOCK (TYPEWRITER CODE IS 0F)
        . IF NOT
445 REM THEN NEW CHAR IS LOWER CASE, AND MUST PRESS SHI
        FT KEY
446 REM TO RELEASE SHIFT LOCK (TYPEWRITER CODE IS 0E).
447 REM SO IF A REG EQUALS 40, INCREMENT C REG TO 0F OT
        HERWISE
448 REM DO NOT. PUT 40 (TIME DELAY FOR SHIFTS) INTO H R
        EG.
449 REM CALL SUBROUTINE "D" TO SHIFT TO LC OR UC. POP B
        C.
450 DATA C5,0E,0E,28,01,0C,26,40,CD,TD,C1
451 REM
452 REM LABEL "N" JUMP TO HERE IF NO SHIFT NEEDED
453 REM MOVE PREVIOUS TYPED CHARACTER FROM "T" TO A REG
        .
454 REM MOVE A REG TO B REG. MOVE C REG (CODE FOR NEXT
455 REM TYPED CHARACTER) TO A REG, AND FROM A REG INTO
        "T".
456 REM COMPARE A REG TO B REG.
460 DATA LN,3A,TT,47,79,32,TT,B8
461 REM
462 REM IF OLD AND NEW TYPE CODES ARE THE SAME, PUT 0A
463 REM (TIME DELAY) INTO H AND CALL "W" FOR ADDITIONAL
464 REM TIME DELAY.
470 DATA 20,05,26,0A,CD,TW
471 REM
472 REM OTHERWISE PUT 19 (TIME DELAY) INTO H AND CALL "
        D"
473 REM TO PRINT CHARACTER. AFTER ALL, ISN'T THIS THE
474 REM PURPOSE OF THE PROGRAM?
475 REM MOVE C REG (STILL HAS CODE FOR NEXT TYPED CHARA
        CTER)
476 REM INTO A REG TO SEE IF IT IS A CODE WHICH REQUIRE
        S
```

Program continues

er port), sets the four low order bits to zero (the AND F0 instruction) and then compares with to 30 hex. If the value is not 30, the program jumps back to 52A5 and repeats until the printer is ready, and the value of the high order four bits at the printer port becomes 30 hex. Once this occurs, the POP AF instruction moves the byte to be sent to the printer in the A register, and the LD (nn),A instruction then moves the contents of the A register into location 37E8, the printer port.

The set of instructions which begins at 5293 has a different purpose. The Electric Pencil instructions state that the program checks once to see if a printer is connected, and if it is not, uses the serial output for all subsequent printing. The instructions move the contents of 37E8 to the A register, and then compare to FF, using the result of that comparison for subsequent jumps. It seemed probable to me that an unconnected printer port would be sensed as all "ones", (that is, FF) so that any other value at that location would indicate that a printer was indeed connected.

Once I realized what these instructions were doing (and it took a fair amount of time), I was able to patch into the new printer drive in the following way: The two instruction sequences where the program moves the contents of the printer port of the A register (Program Listing 2) were simply replaced by sequences which loaded 30 hex into the A register, so the program would behave just as if there were a connected ready printer at all times. (Time delays for printing are all included in the new printer patch.) In addition, the instruction) and then compares this to 30 hex. If the value is not 30, the the A register to the printer port was replaced by a subroutine call to the new printer patch. Program Listing 3 summarizes the instuction for altering the Pencil program and reloading it on the String Floppy wafer.

The Printer Driver

The printer driver included here serves six important func-

```
477 REM ADDITIONAL TIME DELAY (BACKSPACE, LINEFEED).
480 DATA 26,19,CD,TD,79
482 REM
483 REM COMPARES TO LINE FEED CODE. IF SAME A0 STORED
484 REM IN H. CALL W
485 DATA FE,0D,20,05,26,A0,CD,TW
487 REM
488 REM COMPARES TO BACKSPACE CODE. IF SAME 80 STORED
489 REM IN H. CALL W. SOME TYPEWRITERS USE 08 NOT 09.
490 DATA FE,09,20,05,26,80,CD,TW
498 REM
499 REM LABEL U - POPS STACK AND RETURNS TO PENCIL OR B
    ASIC
500 DATA LU,F1,C1,E1,C9
501 REM SUBROUTINE D - CODE FROM C OUTPUT TO TYPEWRITER
502 DATA LD,79,D3,F7
503 REM
509 REM SUBROUTINE W - LOOP TIME DEPENDS ON H REGISTER
510 DATA LW,2E,FF,2B,7C,B5,20,FB,C9
511 REM
520 DATA LK,40:      REM STORES CASE OF LAST TYPED CHAR
521 DATA LF,00:      REM STORES FLAG FOR END OF PAGE
522 DATA LR,00:      REM STORES LINE FEED COUNT
523 DATA LT,00:      REM STORES CODE OF LAST TYPED CHAR
524 REM
525 REM LABEL L - FIRST LOCATION OF LOOKUP TABLE.
526 REM ASCII 00-07
530 DATA LL,00,00,00,00,00,00,00,00
534 REM
535 REM  BS TB          CR      ONLY CR USED HERE
540 DATA 08,09,00,00,00,0D,00,00
544 REM
550 DATA 00,00,00,00,00,00,00,00
560 DATA 00,00,00,00,00,00,00,00
564 REM
565 REM  SP  !  "  #  $  %  &  '
570 DATA 3E,D1,E0,D3,D4,D5,D7,A0
574 REM
575 REM  (  )  *  +  ,  -  .  /
580 DATA D9,D0,D8,DD,1C,9A,1E,9F
584 REM
585 REM  0  1  2  3  4  5  6  7   ONE IS LOWER CASE L
590 DATA 90,AC,92,93,94,95,96,97
594 REM
595 REM  8  9  :  ;  <  =  >  ?
600 DATA 98,99,DB,9B,FB,9D,BB,DF
604 REM
```

Program continues

tions: First, it decodes ASCII output and converts it to the appropriate solenoid code. Second, when shifting is required, it activates the shift lock (for uppercase) or shift (for lowercase) key solenoids before activating the character solenoid. Third, timing loops are provided to allow enough delay between typed characters. A short additional delay occurs when the same key is activated twice in succession. Longer delays are used for carriage return and backspace. Fourth, the driver halts printing at the end of each page (of Pencil text) to allow for paper change. Fifth, rarely used characters ($,#) may be redefined to provide other functions. Sixth, the new driver is completely relocatable to any location in memory.

The first part of the printer driver program provides the logic for stopping at the end of each type page. I discarded the idea of having the driver keep its own line count because of problems resetting the system during partial-page printouts, tables, etc. I eventually realized that the Pencil output consists of multiple consecutive linefeeds without intervening typed characters at the end of every text page. (It actually outputs a space, a line feed and a carriage return.) The new patch is written so that if the carriage returns four times in a row without intervening typed characters, the typewriter will continue to print any subsequent carriage returns and spaces, (so that as many lines as required can be skipped), but will halt printing for paper change before any character is printed. To resume printing at that point, press the space bar. To return to the Pencil program and stop printing, press Break and hold until the cursor reappears on the screen.

The remainder of the program is a modification of the one included with the I/O Pak, and is described in detail in the program remarks. In particular, the program provides extra delay when the same key is typed twice in succession. This feature, which is useful for "non-Selectric" typewriters, allows a 20

to 30 percent increase in typing speed for random text beyond that possible for successive typed characters. In addition, rarely used characters ($, #) may optionally be used for 'lower-case one' and backspace functions respectively. The 'lower-case one' is useful in typewriters in which the top left-hand key is not used for the number 'one'.

Program Details

Two program listings are provided. The first is a BASIC program which uses L. Suter's Macropoke assembler (*80-Microcomputing* April, 1980) to create a completely relocatable machine language printer driver. Once this program is run, it may be saved on Stringy Floppy wafer using the (@SAVEn,start addr,292) command, and used without reassembly. As written, it is useable for BASIC programs, though a number of lines could be deleted, (namely the end-of-page logic) for minimal memory use. It may be used for Electric Pencil text by modifying line 350 as outlined in the program listing.

The second listing is a short loader (which I store as program one on a Stringy Floppy wafer) which first checks to ensure that sufficient memory size is reserved for the printer driver, then loads the printer driver machine code (@LOAD2), and offers the choice of printing options for the $ and # symbols. The timing loop values are then displayed for easy modification if desired. Finally, the Electric Pencil program itself is loaded from the same wafer (@ LOAD3) and automatically begun. The unusual order of the program lines is due to the fact that the Pencil overwrites all of the BASIC program but the beginning when it is loaded.

I save the three programs, loader, machine language printer driver and Pencil on a single 10 foot wafer, and will be happy to provide copies of both BASIC program listings to anyone sending me a 20-foot wafer. I will not make copies of the Electric Pencil. As noted above, Exatron sells the Electric Pencil modified for saving text on wafer for $10 above the price of the cassette version.

I had relatively few problems. The cable between the interface and the I/O Pak is a bit short (4 feet), and limits the positioning of the typewriter. Rochester Data was nice enough to send me an 8-foot cable in exchange for the original one when I mentioned the problem. The only other problem of consequence has been an occasional sticking of the solenoid plungers within the coils. This was completely solved by application of a very small quantity of "Tri-flon" Teflon lubricant to the plunger shafts, and has not recurred.

The Rochester Data I/O Pak is a reliable interface between computer and typewriter, and allows generation of high-quality text at minimal cost, though slower than the more expensive solid face printers.■

```
605 REM   @  A  B  C  D  E  F  G  UPPER CASE
610 DATA D2,E1,E2,E3,E4,E5,E6,E7
614 REM
615 REM   H  I  J  K  L  M  N  O
620 DATA E8,E9,EA,EB,EC,ED,EE,EF
624 REM
625 REM   P  Q  R  S  T  U  V  W
630 DATA F0,F1,F2,F3,F4,F5,F6,F7
634 REM
635 REM   X  Y  Z  (     )        D6 = CENTS, DA = UNDE
   RLINE
640 DATA F8,F9,FA,D9,D6,D0,D6,DA
644 REM
645 REM   @  a  b  c  d  e  f  g   LOWER CASE
650 DATA D6,A1,A2,A3,A4,A5,A6,A7
654 REM
655 REM   h  i  j  k  l  m  n  o
660 DATA A8,A9,AA,AB,AC,AD,AE,AF
664 REM
665 REM   p  q  r  s  t  u  v  w
670 DATA B0,B1,B2,B3,B4,B5,B6,B7
674 REM
675 REM   x  y  z  (  <  )  >
680 DATA B8,B9,BA,D9,FB,D0,BB,00,XX
```

Program Listing 5.

```
10 GOTO50
20 @LOAD3
30 END
40 PRINT "YOU NEED MEMORY SIZE 32475 OR LESS. START AGAIN";END
50 IF PEEK(16561)+256*PEEK(16562)]32475 GOTO40
60 @LOAD2
70 CLS:PRINT "             PENCIL LOADING PROGRAM"
80 PRINT:PRINT
90 PRINT "$ PRINTS AS $ (A) OR 'ONE' (B)?";
100 A$=INKEY$:IF A$="A" THEN PRINT "   $":GOTO130
110 IF A$="B" THEN PRINT "   'ONE'":POKE32676,145:GOTO130
120 GOTO100
130 PRINT:PRINT "# PRINTS AS # (A) OR BACKSPACE (B)?";
140 A$=INKEY$:IF A$="A"THEN PRINT "   #":GOTO180
150 IF A$="B" THEN PRINT "   BACKSPACE":POKE 32675,9:GOTO180
160 GOTO 140
170 REM IF BACKSPACE IS AT TOP RIGHT OF TYPEWRITER, POKE 32675,8
180 PRINT
190 PRINT"                          PRESENT VALUE"
200 PRINT"ENTER Q TO TERMINATE"
210 PRINT"  C - CHARACTER DELAY= ",PEEK(32598)
220 PRINT"  I - REPT CHAR INCR = ",PEEK(32593)
230 PRINT"  B - BACKSPACE DELAY= ",PEEK(32617)
240 PRINT"  R - CARR.RET. DELAY= ",PEEK(32608)
250 PRINT"  S - SHIFT DELAY    = ",PEEK(32576)
260 PRINT
270 PRINT "ENTER DELAY CONSTANT (Q,C,I,B,R,S)  ";
280 A$=INKEY$:IFA$="Q"GOTO20
290 IF A$[]"C"ANDA$[]"B"ANDA$[]"R"ANDA$[]"S"ANDA$[]"I"GOTO280
300 PRINT A$
310 INPUT"ENTER NEW VALUE, THEN PRESS 'ENTER'";N
320 IF A$="C"AND N]20 POKE 32598,N:GOTO200
330 IF A$="I"AND N]0 POKE 32593,N:GOTO200
340 IF A$="B"AND N]60 POKE 32617,N:GOTO200
350 IF A$="R"AND N]60 POKE 32608,N:GOTO200
360 IF A$="S"AND N]20 POKE 32576,N:GOTO200
370 GOTO200
```

Some suggestions on using READ, DATA and RESTORE.

On Embedding Data

John D. Adams
13126 Tripoli Ave.
Sylmar, CA 91342

Computer programs operate on data, and that data can be broadly divided into two categories. Some data will not change from one program run to the next; such a value is called a constant. Other data may change with each program run, and these values are referred to as variables. For example, we find the area of a circle by multiplying the square of the radius by π. In this operation, π is a constant and the measure of the radius is a variable.

Somewhere in the program we must enter the data into the computer. To find the area of a circle we could do it this way: 10 INPUT"ENTER PI";PI:INPUT"ENTER RADIUS";R. This line will load the data, but is somewhat inefficient. Each time we run it, π has to be re-entered.

A better idea would be to put the value of π into the program itself, or embed that value in the program. Such a line would look like this: 10 PI=3.14159:INPUT"ENTER RADIUS";R. Now when the program is loaded, so is the value for π. Embedding constants in the program is more practical than entering them over and over.

Efficient Embedding

The READ, DATA and RESTORE instructions are very efficient at embedding constants. Used in conjunction with FOR—NEXT loops, constant data can be loaded into memory locations rapidly and precisely.

Assume that we need to store the days of the week in locations A$(1) through A$(7). It could be done this way:

```
10 A$(1)="MONDAY"
20 A$(2)="TUESDAY"
30 A$(3)="WEDNESDAY"
Etc.
```

This will load the data, but the following will do the same:

```
10 FOR X=1 TO 7:READ A$(X):NEXT
20 DATA SUNDAY,MONDAY,TUESDAY,WEDNESDAY,
THURSDAY,FRIDAY,SATURDAY
```

The latter method is more efficient, both in time and in byte space. As the amount of data increases the savings become more substantial.

The READ A$ instruction is used in conjunction with the data instruction, and they actually form a routine. When the computer encounters a read instruction it looks for the first data line and then takes the first available item of data in that line and loads it into the indicated memory location. More than one location can be loaded with a single read instruction and more than one item of data can be written into a single line.

Memory locations and data are separated by commas, and a one-to-one correspondance exists between memory location and data item. The first data item will be read into the first listed location, the second item into the second location, and so forth.

Every time the TRS-80 loads an item of data it marks off that item so it will not be used again. The item is not lost forever, however, and can be restored.

When there is too much data to be listed in a single line, another line with a greater number can be started, and it will be read after the data in the first line is exhausted.

Requirements

There are some requirements which must be observed when using the READ-DATA instructions. First is the string/non-string problem. Trying to use the line READ A:DATA MARCH will get you an error message. A is a *value* location and will not accept string data: You must match up string locations with string data and vice versa. Numbers may be loaded into string locations, such as in READ A$:DATA 1980, but such data are then loaded as symbols and not as values. Asking the computer to PRINT A$+A$ will not return 3960; 19801980 is returned, as a concatenation has been indicated and not a sum.

Precision presents another problem. Run these lines:

```
10 READ A:DATA .3333333333333333
20 B#=A:PRINT B#
```

What happened to the nice string of threes? The data is a double precision figure and it was read into a single precision memory location. We then shifted back to double precision in the B# location and destroyed accuracy. If we try to get double precision numbers from single precision locations, anything after the sixth digit may be nonsense. When dealing with different precisions it pays to be scrupulous about using type declarations on variable names.

Also be careful about the number of items involved. Look at the following line: 10 READ A,B,C,D,E:DATA 1,2,3,4. Running this line will return an ?OD, or out of data, message. The computer cannot read five items of data when only four are given. There must be an item of data for each specified memory location. We can, on the other hand, have more data items than read locations, and this can be used to advantage if the data you are using needs to be updated periodically.

```
10 FOR X=1 TO 2
20 FOR Y=1 TO 5
30 READ A(Y):PRINT A(Y);
40 NEXT:NEXT
50 DATA 0,1,2,3,4,5,6,7,8,9
```

On the first complete cycle of the Y loop,

the numbers 0, 1, 2, 3 and 4 are loaded into the indicated locations and then printed. On the second cycle of the Y loop numbers five to nine are written over in the old locations and the printout finishes with all ten digits printed. When using these instructions, it is advisable to match locations and items with attention to type, precision and number of items.

The comma is used to denote the end of a data item so that more than one item can be listed in a single line. This can pose a problem if commas are needed inside a single item of data. The line: 10 READ A$:DATA SMITH, WILLIAM is not going to work as expected. The program will not crash, but all that will be loaded into A$ is SMITH. The comma is the culprit, denoting the end of the first data item.

Since the colon is used to define multiple line statements, it is also off limits, as are leading spaces, although trailing spaces may be included. Run the following: 10 READ A$,B$:DATAONE, TWO. Even though a space was included before the word two, the printout reads ONETWO.

There are several ways of getting around these limitations and they are illustrated in the following lines.

```
10 READ F$,M$,L$:DATA MARY, T.,BROWN
20 READ CM$,CO$,SP$:DATA ",",":"," "
30 PRINT L$;CM$;SP$;F$;SP$;M$;CO$
40 X$ = " ":Y$ = ":":Z$ = ","
50 PRINT L$;Z$;X$;F$;X$;M$;Y$
60  PRINT  L$;CHR$(44);CHR$(32);F$;CHR$(32);
M$;CHR$(58)
```

Line 10 loads MARY into F$, T. into M$ and BROWN into L$. The first method is illustrated in lines 20 and 30. Although illegal characters cannot be loaded as parts of data items, they may be loaded if enclosed in quotes. Line 20 loads a comma into CM$, a colon into CO$ and a space into SP$. Printout is accomplished by line 30.

Lines 40 and 50 show another method. Here the comma, colon and space are loaded into separate string locations by assignment. Printout is done by line 50. The CHR$ instruction may also be used as demonstrated in line 60. Tricky characters are easily printed using this system. As the ASCII number for the space is 32, for the comma 44 and for the colon 58, line 60 accomplishes the desired effect.

It is possible that after data has been loaded and marked off once, it may be needed again, and the restore instruction is used for that purpose. Its use allows data that has been read to be read again starting with the first item. Note that *all* data will be restored, and you cannot be selective about what you need without special programming. The following lines illustrate the use of the restore statement:

```
10 FOR X = 1 TO 2
20 FOR Y = 1 TO 5
30 READ A(Y):PRINT A(Y);
40 NEXT
50 RESTORE
60 NEXT
70 DATA 1,2,3,4,5
```

On the first cycle through loop Y all the data was used, but as the computer comes out of the loop and hits the restore instruction in line 50, all the data is again available so that the second pass of the Y loop can use it again. If line 50 is deleted the program will crash, as there is not enough data available. Placement and use of the restore instruction requires practice. These three statements are indispensible in programs in which a lot of constant data is needed. ■

```
10 FOR X=1 TO 2
20 FOR Y=1 TO 5
30 READ A(Y):PRINT A(Y);
40 NEXT:NEXT
50 DATA 0,1,2,3,4,5,6,7,8,9
```

Program Listing 1.

```
10 READ F$,M$,L$:DATA MARY,T.,BROWN
20 READ CM$,CO$,SP$:DATA ",",":"," "
30 PRINT L$;CM$;SP$;F$;SP$;M$;CO$
40 X$=" ":Y$=":":Z$=","
50 PRINT L$;Z$;X$;F$;X$;M$;Y$
60 PRINT L$;CHR$(44);CHR$(32);F$;CHR$(32);M$;CHR$(58)
```

Program Listing 2.

```
10 FOR X=1 TO 2
20 FOR Y=1 TO 5
30 READ A(Y):PRINT A(Y);
40 NEXT
50 RESTORE
60 NEXT
70 DATA 1,2,3,4,5
```

Program Listing 3.

No pre-payment penalty on your loan? Good. Save some interest charges by paying early: all you need is this program and some money.

The Rule of 78s

R. L. Conhaim
15506 Kiamichi Rd., Apt. 1
Apple Valley, CA 92307

In the old days, before Uncle Sam decided to protect us from ourselves, life was simple. It didn't take much schooling to be able to figure interest rates.

In those days, if you borrowed $2,500 to buy a car, and the finance charge was $450 for the three years of the loan, the interest rate calculation was easy. Divide the finance charge by the three years of the loan and you have $150. Divide that by $2,500 and you get .06, which is the decimal equivalent of six percent. Right? Not any more.

The six percent was right for the old days and was called the add-on rate. It was very obvious that if some finance company offered you a five and a half percent loan it was a better deal than six percent. But, some folks in Washington thought we were being ripped off, so they dreamed up something called Truth in Lending and Federal Reserve Regulation Z.

Annual Percentage Rate

The six percent add-on rate, they told us, didn't really tell the story because it was based on the total amount of the loan. What we must now be quoted, according to Regulation Z, is the Annual Percentage Rate or APR. This is figured on the upaid balance, month by month. So our six percent loan turns out to be more like 11.08 percent. But, who does the figuring? Therein lies the rub.

APR calculation is a lengthy and involved procedure that could take hours to figure with pencil, paper and log tables. Even the abacus wizards would have touble. It's no wonder that computer programs you see covering direct reduction loans tell you how to figure everything but APR.

Realizing the difficulty of calculating APR, the Federal Reserve Board obligingly offered a book of Annual Percentage Rate Tables. But, even if you had the book, you still had to do some figuring, and strict accuracy often made it necessary to interpolate between columns of the tables. The chance for mistake was great.

If you got curious, you could send away for Supplement I to Regulation Z, which contains the general equations for figuring APR. Taking our orignal example, and assuming a monthly payment of $81.94 ($2950/36), it could be solved like this:

$$2500 = \frac{81.94}{(1+i)^1} + \frac{81.94}{(1+i)^2} + \ldots + \frac{81.94}{(1+i)^{36}}$$

To use it you plug in a guess for i, go through the whole procedure and see how close your guess was to computing 2500. If you're not close enough, you change the value of i and go through the procedure again, repeating as often as necessary so the final APR value gets within ¼ of one percent of the correct figure.

In 1969 when the regulation went into effect you had to have a computer to solve such a problem. Fortunately, programmable calculators, and pre-programmed financial calculators came along to fill the void. But they are relatively slow, and all the methods used to solve for APR are iterative (repeating) procedures that require you to make an initial guess. The speed of the process is dependent on the accuracy of your guess, and typical times for calculator solutions vary from 15 to 45 seconds.

The Hewlett-Packard HP-80, a financial calculator, figured APR from the add-on rate, using this formula:

$$\frac{n}{1 + \frac{n}{12}r} = \frac{1-(1+i)^{-n}}{i}$$

where r is the decimal add-on rate and APR = 1200 i.

The math book that applied to

```
10 CLS
20 PRINT STRING$(15,"-");"APR CALCULATION";STRING$(15,"-")
30 PRINT
40 INPUT "ENTER AMOUNT BORROWED";PV
50 INPUT "ENTER PAYMENT AMOUNT $";PM
60 INPUT "ENTER NUMBER OF MONTHS";N
70 I= (1/(PV/PM))-((1/(N[2))*(PV/PM))
80 A = (PV/PMT)*I
90 B = (1+I)[-N
100 C= (1+(N/(1+(1/I))))
110 D= A-(1-B)
120 E= (C*B)-1
130 F= (D/(E/I))
140 G= F[2
150 I = I + F
160 IF G=<1E-12 THEN 180 ELSE 80
180 CLS:PRINT:PRINT:PRINT
190 PRINT "THE ANNUAL PERCENTAGE RATE IS ";
200 PRINT USING "##.####%";I*1200
210 PRINT
220 X=0
230 INPUT "DO YOU WANT ANOTHER CALCULATION (Y/N)";J$
240 IF J$ = "N" GOTO 270 ELSE 10
270 END
```

Program Listing 1. Left bracket is an up arrow.

✓See List of Advertisers on page 306

the HP-55 programmable calculator used a slightly different formula, where you needed to know the principal, payment and number of periods. It looked like this:

$$i_{k+1} = i_k - \frac{f(i)}{f'(i)} \text{ where}$$

$$f(i) = \frac{1-(1+i)^{-n}}{i} - \frac{\text{Principal}}{\text{Payment}} \text{ and}$$

$f'(i)$ is the first derivative of $f(i)$

I can hear you saying, "Stop with the math, already!" The point of all this is that APR calculation is difficult, and that's why you don't see it in published programs. But, the truth of the matter is that it can be easily solved by your computer.

Short and Sweet

The TRS-80 BASIC program in Listing 1 is short and sweet. The calculations have been broken down into easily digestible portions. The calculations keep repeating until the change from one answer to the next is equal to or less than 10^{-12}. That's far better accuracy than the required ¼ of one percent required by Regulation Z.

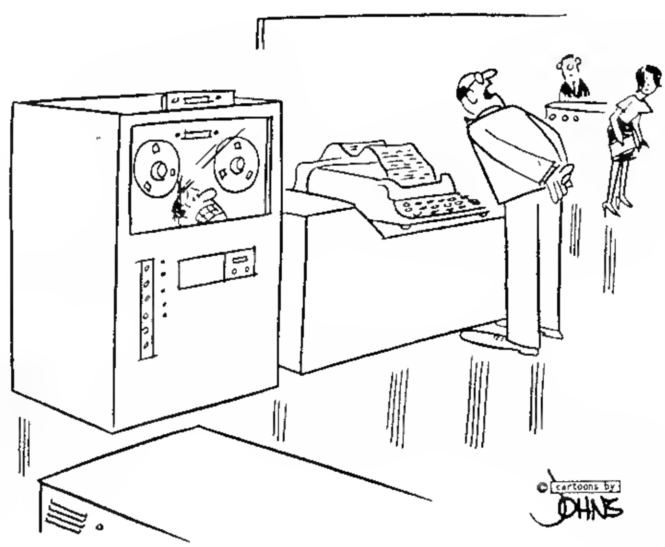
What about speed and accuracy? Take a look at a similar problem and its results:

HP-55	10.3165%	30 seconds
T.I Bus. Analyst	10.3165%	10 seconds
TRS-80	10.3156%	1½ seconds

In the case of both calculators the result was a monthly percentage rate which you had to multiply times 12 to get APR. The TRS-80 output is true APR.

It is interesting to note that using double precision with the TRS-80 accomplishes nothing of importance. If you're interested in other values of the direct reduction loan formulas, you can find them in many books and financial software.

Here, then, is one more example of something your computer can do better and faster than other methods, with comparable accuracy. Like the financial calculators, you don't even have to make a guess. The program does that for you.■



" WHO'S THE CLOWN WHO WROTE 'I'M BEING HELD IN THIS COMPUTER AGAINST MY WILL!'? "

See List of Advertisers on page 306

80 microcomputing™ bookshelf

—INTRODUCTORY—

new

● **YOUR FIRST COMPUTER**—BK1191—by Rodnay Zaks. Whether you are using a computer, thinking about using one or considering purchasing one, this book is indispensable. It explains what a computer system is, what it can do, how it works and how to select various components and peripheral units. It is written in everyday language and contains invaluable information for the novice and the experienced programmer. (The first edition of this book was published under the title "An Introduction to Personal and Business Computing".) $7.95*

● **SOME OF THE BEST FROM KILOBAUD MICROCOMPUTING**—BK7311—A collection of the best articles that have appeared in Kilobaud MICROCOMPUTING. Included is material on the TRS-80 and PET systems, CP/M, the 8080/8085/Z-80 chips, the ASR-33 terminal. Data base management, word processing, text editors and file structures are covered too. Programming techniques and hard-core hardware construction projects for modems, high speed cassette interfaces and TVTs are also included in this large format, 200 plus page edition. $10.95*

● **HOBBY COMPUTERS ARE HERE!**—BK7322—If you want to come up to speed on how computers work... hardware and software... this is an excellent book. It starts with fundamentals and explains the circuits, and the basics of programming, along with a couple of TVT construction projects, ASCII-Baudot, etc. This book has the highest recommendations as a teaching aid. $4.95*

● **UNDERSTANDING AND PROGRAMMING MICROCOMPUTERS**—BK7382—A valuable addition to your computing library. This two-part text includes the best articles that have appeared in 73 and Kilobaud Microcomputing magazines on the hardware and software aspects of microcomputing. Well-known authors and well-structured text helps the reader get involved. $10.95*

● **THE NEW HOBBY COMPUTERS**—BK7340—This book takes it from where "HOBBY COMPUTERS ARE HERE!" leaves off, with chapters on Large Scale Integration, how to choose a microprocessor chip, an introduction to programming, low cost I/O for a computer, computer arithmetic, checking memory boards... and much, much more! Don't miss this tremendous value! Only $4.95*

— INTRODUCTION — TO MICROCOMPUTERS (VOL. 0–III)

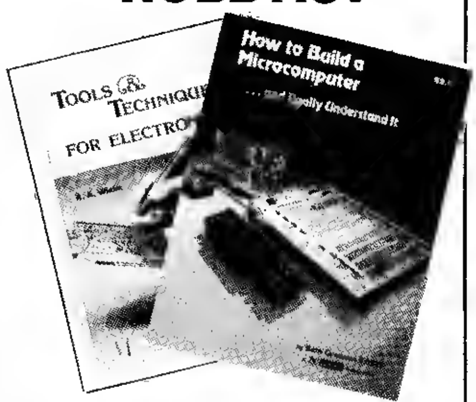
● **AN INTRODUCTION TO MICROCOMPUTERS, VOL. 0**—BK1130—The Beginner's Book—Written for readers who know nothing about computers—for those who have an interest in how to use computers—and for everyone else who must live with computers and should know a little about them. The first in a series of 4 volumes, this book will explain how computers work and what they can do. Computers have become an integral part of life and society. During any given day you are affected by computers, so start learning more about them with Volume 0. $7.95*

● **VOL. I**—BK1030—**2nd Edition completely revised.** Dedicated to the basic concepts of microcomputers and hardware theory. The purpose of Volume I is to give you a thorough understanding of what microcomputers are. From basic concepts (which are covered in detail), Volume I builds the necessary components of a microcomputer system. This book highlights the difference between minicomputers and microcomputers. $12.99*

● **VOL. II**—BK1040 (with binder)—Contains descriptions of individual microprocessors and support devices used only with the parent microprocessor. Volume II describes all available chips. $31.99*

● **VOL. III**—BK1133 (with binder)—Contains descriptions of all support devices that can be used with any microprocessor. $21.99*

FOR THE ELECTRONICS — HOBBYIST —



● **TOOLS & TECHNIQUES FOR ELECTRONICS**—BK7348—by A. A. Wicks is an easy-to-understand book written for the beginning kit builder as well as the experienced hobbyist. It has numerous pictures and descriptions of the safe and correct ways to use basic and specialized tools for electronic projects as well as specialized metal working tools and the chemical aids which are used in repair shops. $4.95*

● **HOW TO BUILD A MICROCOMPUTER—AND REALLY UNDERSTAND IT**—BK7325—by Sam Creason. The electronics hobbyist who wants to build his own microcomputer system now has a practical "How-To" guidebook. This book is a combination technical manual and programming guide that takes the hobbyist step-by-step through the design, construction, testing and debugging of a complete microcomputer system. Must reading for anyone desiring a true understanding of small computer systems. $9.95*

—SPECIAL INTERESTS—

new

● **THE CP/M HANDBOOK (with MP/M)**—BK1187—by Rodnay Zaks. A complete guide and reference handbook for CP/M—the industry standard in operating systems. Step-by-step instruction for everything from turning on the system and inserting the diskette to correct user discipline and remedial action for problem situations. This also includes a complete discussion of all versions of CP/M up to and including 2.2, MP/M and CDOS. $14.95*.

● **INTRODUCTION TO TRS-80 GRAPHICS**—BK1180—by Don Inman. Dissatisfied with your Level I or Level II manual's coverage of graphics capabilities? This well-structured book (suitable for classroom use) is ideal for those who want to use all the graphics capabilities built into the TRS-80. A tutorial method is used with many demonstrations. It is based on the Level I, but all material is suitable for Level II use. $8.95*

● **TRS-80 DISK AND OTHER MYSTERIES**—BK1181—by Harvard C. Pennington. This is the definitive work on the TRS-80 disk system. It is full of detailed "How to" information with examples, samples and in-depth explanations suitable for beginners and professionals alike. The recovery of one lost file is worth the price alone. $22.50*

● **MICROSOFT BASIC DECODED AND OTHER MYSTERIES**—BK1186—by James Farvour. From the company that brought you *TRS-80 DISK AND OTHER MYSTERIES!* Contains more than 6500 lines of comments for the disassembled Level II ROMs, six additional chapters describing every BASIC subroutine, with assembly language routines showing how to use them. Flowcharts for all major routines give the reader a real insight into how the interpreter works. $29.50.

● **MICROPROCESSOR INTERFACING TECHNIQUES**—BK1037—by Austin Lesea & Rodnay Zaks—will teach you how to interconnect a complete system and interface it to all the usual peripherals. It covers hardware and software skills and techniques, including the use and design of model buses such as the IEEE 488 or S-100. $15.95*

*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to 80 Microcomputing Bookshelf ● Peterborough NH 03458. Be sure to include check or detailed credit card information.

No C.O.D. orders accepted. All above add $1.00 handling. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write Customer Service at the above address.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

PRICES SUBJECT TO CHANGE WITHOUT NOTICE

Why use their flexible discs:

Ampex, Athana, BASF, Caelus, Control Data, Dysan, IBM, Inmac, K-Line, Maxell, Nashua, Scotch, Shugart, Syncom, 3M, Verbatim or Wabash

when you could be using

MEMOREX

for as low as $1.99 each?

Find the flexible disc you're now using on our cross reference list... then write down the equivalent Memorex part number you should be ordering.

Product Family	Product Description	Memorex Part Number (3201-)	CE quant. 100 price per disc ($)	Ampex	Athana	BASF	Caelus	Dysan	IBM	K Line	Maxell	Nashua	Scotch 3M	Shugart	Syncom	Verbatim	Wabash	Inmac	Control Data
Flexible Disc 1s, Single Headed Drives, Single-Density Media	IBM Compatible (128 B/S, 26 sectors)	3060	2.19	SFD-11111D	473071	53428	CM-F11	800506	2305830	40013	FD1-128	FD-1	740-D	S/A 100	15002	FD34-1000	F111111X	7870-A	421602
	IBM Compatible (128 B/S, 26 sectors) w/ W P H	3062	2.24	–	–	–	–	–	–	–	–	–	740-D	–	–	–	–	–	–
	IBM Compatible (128 B/S, 26 sectors) w/ W P H & Hub ring	3064	2.55	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	IBM Compatible (128 B/S, 26 sectors) REVERSIBLE	1729	3.36	SFD-11511D	473072	54431	–	–	–	40019	–	FD-2	74D/2-D	–	15150	FF34-2000	F171111X	7860-K	–
	IBM System 6 Compatible	3068	2.19	–	473017	54561	–	800509	1669959	40014	–	–	740-0 056	–	15003	FD60-1000	F118111X	–	–
	IBM Compatible (256 B/S, 15 sectors)	3109	2.19	SFD-11121D	473073	–	–	800584	2305845	40040	–	–	74D-3600	–	15005	FD36-1000	F112111X	7881-K	–
	IBM Compatible (512 B/S, 8 sectors)	3110	2.19		473074	–	–	800585	1669954	40044	–	–	–	–	15004	FD60-1000	F113111X	7889-K	–
	Shugart Compatible, 32 hard sector	3015	2.19	SFD-21101D	470901	53802	CM-F21	101/1	–	40016	FH1-32	FD-122	740-32	S/A-101	15025	FD32-1000	–	7890-K	421322
	Shugart Compatible, 32 hard sector REVERSIBLE	3025	3.35	SFD-21301D	–	–	–	–	–	40017	–	–	74D/2-32	–	15151	FF32-2000	–	7880-K	–
	Wang Compatible, 32 hard sector w/Hub ring	3067	2.50	–	–	54491	–	–	–	–	–	–	740-32RM	–	–	–	F374411X	–	–
	CPT 8000 Compatible	3045	2.79	–	–	–	–	–	–	–	–	–	–	–	15228	–	–	–	–
Flexible Disc 1d, Single-Headed Drives, Double-Density Media	IBM Compatible (128 B/S, 26 sectors)	3090	2.95	SFD-121010	474011	54568	–	3740/1D	–	40047	FD1-128/M2100	FD-1D	741-0	–	–	FD34-8000	F131111X	7857-K	423002
	Soft Sector (128 B/S, 26 sectors) REVERSIBLE	3093	3.99	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	Shugart Compatible, 32 hard sector	3091	2.95	SFD-221010	470801	54596	–	101/1D	–	40024	FH1-320	–	741-32	S/A-103	15075	FD32-8000	F334410X	7887-K	423322
	Shugart Compatible, 32 hard sector REVERSIBLE	3094	3.99	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	Wang Compatible, 32 hard sector w/ Hub ring	3088	3.20	–	–	–	–	–	–	–	–	–	–	–	–	–	F224411X	–	–
Flexible Disc 2s, Double Headed Drives, Single Density Media	Soft Sector (Unformatted)	3101	3.84	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	Soft Sector (128 B/S, 26 sectors)	3113	3.84	–	–	54425	–	800814	1766870	–	–	–	–	S/A-150	15153	FD10-4026	F121111X	–	–
	Soft Sector (256 B/S, 15 sectors)	3106	3.84	–	473477	54226	–	800813	2736700	40043	FD2-2560	–	742-0	–	15154	FD10-4019	F122111X	7656-K	424812
	32 Hard Sector	3108	3.84	–	–	–	–	–	–	–	FH2-32	–	–	–	–	–	–	–	–
Flexible Disc 2d, Double-Headed Drives, Double-Density Media	Soft Sector (Unformatted)	3102	3.49	–	473465	–	–	DY15D	–	40028	FD2-XDM	FD-2D	743-0	–	15103	DD34-4001	–	–	425002
	Soft Sector (128 B/S, 26 sectors)	3115	3.49	–	–	–	–	–	–	–	–	–	–	S/A-150	–	–	–	–	–
	Soft Sector (256 B/S, 26 sectors)	3103	3.49	–	473471	54329	–	800817	1766872	40019	FD2-2560	–	743-0/256	–	15101	DD34-4026	F144111X	7858-K	425602
	Soft Sector (512 B/S, 15 sectors)	3114	3.49	–	473472	54479	–	800818	1669044	40039	–	–	743-0/512	–	15100	DD34-4019	F145111X	–	425612
	Soft Sector (1024 B/S, 8 sectors)	3104	3.49	–	473473	54485	–	800819	1669045	40020	–	–	743-0/1024	–	15102	DD34-4008	F147111X	7859-K	425622
	32 Hard Sector	3105	3.49	SFD-321010	470851	–	–	101/2D	–	40021	FH2-320	–	743-32	S/A-151	15125	DD32-4000	F344411X	7881-K	425322
	Burroughs B-80 Compatible 32 Hard Sector	3092	3.49	–	–	–	–	–	–	–	–	–	–	–	–	–	F344811X	–	–
	Soft Sector (1024 B/S, 8 sectors) w/ Hub Ring	3118	3.75	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Flexible Disc FD, Memorex 651 or Equiv Drive Compatible	FD VI (Vinyl Jacket)	30712003	2.85	–	470651	–	CM-831	FDIV	–	40002	–	FD-165	511-0	–	15028	FD65-1000	F651411X	7910	–
Mini Flexible Disc 1s, 5¼ Single-Headed Drives, Single-Density Media	Soft Sector (Unformatted)	3401	1.99	–	475001	54256	–	104/1	–	40500	MD1	MD 1	744-0	S/A-104	15300	MD525-01	M111A211X	7897	441002
	10 Hard Sector	3403	1.99	–	475010	54257	–	107/1	–	40501	–	MD 1-10	744-10	S/A-107	15325	MD525-10	M141A211X	7896	441102
	16 Hard Sector	3405	1.99	–	475016	54258	–	105/1	–	40502	MH1	MD 1-16	744-16	S/A-105	15326	MD525-16	M151A211X	7899	441162
	Soft Sector (Unformatted) W/Hub Ring	3431	2.19	–	–	–	–	–	–	–	–	–	–	–	–	MD525-01	–	–	–
	10 Hard Sector w/Hub Ring	3433	2.19	–	–	–	–	–	–	–	–	–	–	–	–	MD525-10	–	–	–
	16 Hard Sector w/Hub Ring	3435	2.19	–	–	–	–	–	–	–	–	–	–	–	–	MD525-16	–	–	–
Mini Flexible Disc 1d, 5¼ Single-Headed Drives, Double-Density Media	Soft Sector	3417	2.24	–	–	54646	–	104/1D	–	–	–	–	–	–	–	MD540-01	–	–	–
	10 Hard Sector	3415	2.24	–	–	54649	–	107/1D	–	–	–	–	–	–	–	MD540-10	–	–	–
	16 Hard Sector	3419	2.24	–	–	54652	–	105/1D	–	–	–	–	–	–	–	MD540-16	–	–	–
Mini Flexible Disc 2d, 5¼ Double Headed Drives, Double Density Media	Soft Sector	3421	2.74	–	–	54824	–	104/2D	–	–	MD2-0	–	745-0	S/A-154	–	MD550-01	–	–	–
	10 Hard Sector	3423	2.74	–	–	54827	–	107/2D	–	–	–	–	745-10	S/A-157	–	MD550-10	–	–	–
	16 Hard Sector	3425	2.74	–	–	54630	–	105/2D	–	–	MH2-0	–	745-16	S/A-145	–	MD550-16	–	–	–

Memorex Flexible Discs...The Ultimate in Memory Excellence

Quality
Memorex means quality products that you can depend on. Quality control at Memorex means starting with the best materials available. Continual surveillance throughout the entire manufacturing process. The benefit of Memorex's years of experience in magnetic media production, resulting, for instance, in proprietary coating formulations. The most sophisticated testing procedures you'll find anywhere in the business.

100 Percent Error Free
Each and every Memorex Flexible Disc is certified to be 100 percent error free. *Each* track of *each* flexible disc is tested, individually, to Memorex's stringent standards of excellence. They test signal amplitude, resolution, low-pass modulation, overwrite, missing pulse error and extra pulse error. They are torque-tested, and competitively tested on drives available from almost every major drive manufacturer in the industry including drives that Memorex manufactures. Rigid quality audits are built into every step of the manufacturing process and stringent testing result in a standard of excellence that assures you, our customer, of a quality product designed for increased data reliability and consistent top performance.

Customer-Oriented Packaging
Memorex's commitment to excellence does not stop with a quality product. They are proud of their flexible discs and they package them with pride. Both their packaging and their labeling have been designed with your ease of identification and use in mind. The desk-top box containing ten discs is convenient for filing and storage. Both box labels and jacket labels provide full information on compatibility, density, sectoring, and record length. Envelopes with multi-language care and handling instructions and color-coded removable labels are included. A write-protect feature is available to provide data security.

Full One Year Warranty — Your Assurance of Quality
Memorex Flexible Discs will be replaced free of charge by Memorex if they are found to be defective in materials or workmanship within one year of the date of purchase. Other than replacement, Memorex will not be responsible for any damages or losses (including consequential damages) caused by the use of Memorex Flexible Discs.

Quantity Discounts Available
Memorex Flexible Discs are packed 10 discs to a carton and 10 cartons to a case. Please order only in increments of 100 units for quantity 100 pricing. We are also willing to accommodate your smaller orders. Quantities less than 100 units are available in increments of 10 units at a 10% surcharge. **Quantity discounts** are also available. Order 500 or more discs at the same time and deduct 1%; 1,000 or more saves you 2%; 2,000 or more saves you 3%; 5,000 or more saves you 5%; 10,000 or more saves you 7%; 25,000 or more saves you 8%; 50,000 or more saves you 9% and 100,000 or more discs earns you a 10% discount off our super low quantity 100 price. Almost all Memorex Flexible Discs are immediately available from **CE**. Our warehouse facilities are equipped to help us get you the quality product you need, when you need it. If you need further assistance to find the flexible disc that's right for you, call the Memorex compatibility hotline. Dial 800-538-8080 and ask for the *flexible disc hotline* extension 0997. In California dial 800-672-3525 extension 0997.

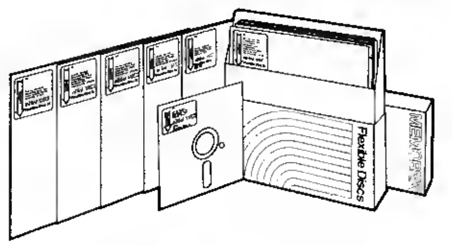
Buy with Confidence
To get the fastest delivery from **CE** of your Memorex Flexible Discs, send or phone your order directly to our Computer Products Division. Be sure to calculate your price using the **CE** prices in this ad. Michigan residents please add 4% sales tax. Written purchase orders are accepted from approved government agencies and most well rated firms at a 10% surcharge for net 10 billing. All sales are subject to availability. All sales are final. Prices, terms and specifications are subject to change without notice. Out of stock items will be placed on backorder automatically unless **CE** is instructed differently. International orders are invited with a $20.00 surcharge for special handling in addition to shipping charges. All shipments are F.O.B. Ann Arbor, Michigan. No COD's please. Non-certified and foreign checks require bank clearance.
Mail orders to: **Communications Electronics**, Box 1002, Ann Arbor, Michigan 48106 U.S.A. Add $8.00 per case or partial-case of 100 8-inch discs or $6.00 per case of 100 5¼-inch mini-discs for U.P.S. *ground* shipping and handling in the continental U.S.A. If you have a Master Charge or Visa card, you may call anytime and place a credit card order. **Order toll-free** in the United States. Call anytime 800-521-4414. If you are outside the U.S. or in Michigan, dial 313-994-4444. **Dealer Inquiries Invited.** All order lines at **Communications Electronics** are staffed 24 hours.

MEMBER

VISA

master charge

Order Toll-Free! (800) 521-4414



For Data Reliability—Memorex Flexible Discs

COMMUNICATIONS ELECTRONICS™ ✓263

Computer Products Division

854 Phoenix ☐ Box 1002 ☐ Ann Arbor, Michigan 48106 U.S.A.
Call TOLL-FREE (800) 521-4414 or outside U.S.A. (313) 994-4444

Fully TRS-80® Compatible

MPI B/51 DISK DRIVE

FREE CATALOG
Hundreds of other computer products at major savings. Cut out this coupon and send with your name and address.

$321.00
We pay UPS charges in Continental US

In Stock! Limited Quantities.

Includes: Case and Power Supply

Fully tested
Guaranteed for 90 days!

Here's why the MPI/B-51 is the drive for success:

- 40 tracks
- 5 ms track-to-track
- Auto-eject
- Hi-Temp stability
- Fully-closable door
- Speed constant <1½%
- Double density head
- Optical sensors—no switches
- 102K per disk

ADDS MORE POWER TO YOUR SYSTEM

Save time . . . Order by phone

Toll free: 1-800-621-3229
IN ILLINOIS CALL: 312-987-1024

TRS-80™ TANDY CORP

Other Money Savings Opportunities Order by Phone or Mail

CABLES
- 2 Drive Cable $29.00
- 4 Drive Cable $39.00

Diskettes
- Verbatim . . . . 10 for $31.00
- Dysan . . . . 10 for $41.00
- Plastic File Box $3.95

Operating Systems
- TRSDOS 2.3 . . . . $14.95
- 40 Track Patch $9.95
- NEWDOS + 40 track $99.00
- Newdos 80 . . . . $139.00
- TRSDOS Manual $5.95

TRS-8Ø
- 16K Model III $699.00
- Expansion Interface ØK $274.00

Printers
- Centronics 737-1 . . . . $815.00
- IDS 460 . . . . $1219.00
- Epson MX 80 . . . . . . $550.00
- Okidata M80 . . . . . . $679.00
- Microline 82 . . . . . $862.00
- Microline 83 . . . . $1099.00
- Okidata prices include tractors
- NEC Spinwriter 5530 (freight collect) $2579.00

16K Memory Kits
- Prime NEC 200ns dynamic RAM. Comes with complete instructions $39.00
- MPI Service Manual $3.00
- MPI Engineering Manual $30.00

MIDWEST COMPUTER PERIPHERALS
1467 S. Michigan Ave. Chicago, IL 60605

✓8 030181

Midwest Computer Peripherals

1467 S. MICHIGAN • CHICAGO, ILLINOIS 60605

master charge · VISA · AMERICAN EXPRESS Card ACCEPTED!

030181

Quantity	Description	$ each	Total
		6% Ill. Tax	
		TOTAL	

(Min. Order $10.00)

☐ Check enclosed

Bill my ☐ Visa ☐ Am Ex ☐ Master Charge

Acc. No. ______________ Exp. ______

☐ Please send catalog

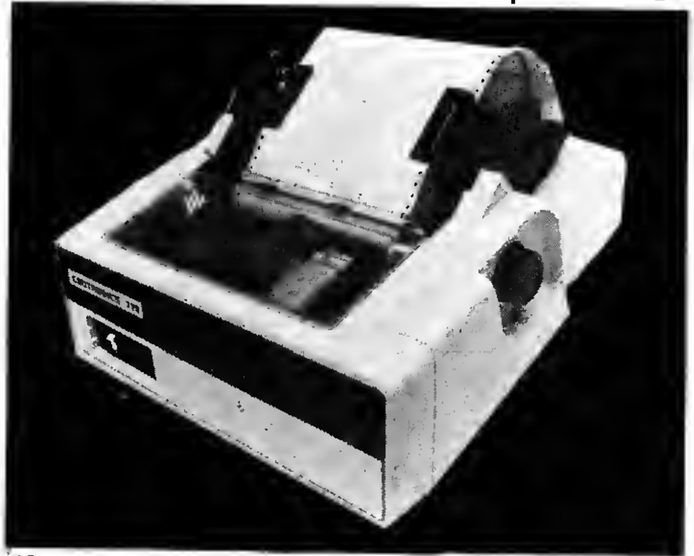
Name ______________

Address ______________

City ______________ State ______ Zip ______

Now save even more with factory rebates on Centronics® printers for the TRS-80®

CENTRONICS 779
Same as Radio Shack Line Printer I
LIST PRICE $1350 **OUR PRICE $799**



(Ship freight collect)

CENTRONICS 737
Featuring Correspondence, Quality and Proportional Spacing *LIST $995*
OUR PRICE $799.
less rebate of **150.**
net cost: ***$649***
(add $7.50 for shipping)

Receive factory Rebates from Centronics with coupon from **MMM**

CENTRONICS 730
Same as Radio Shack Line Printer II

LIST $795
OUR PRICE $599.
less rebate of **100.**
net cost: ***$499***
(add $7.50 for shipping)

TM TRS-80 is a trademark of Radio Shack

Disc Drive Values for TRS-80
Your Choice
5¼" Disk Drives
Shugart, MPI-51, or Tandon
w/case & Power Supply

$499 VALUE
MMM PRICE ONLY

$329
2 for $638

24

Mini Micro Mart, Inc.
1618 James Street, Syracuse, NY 13203
(315) 422-4467 TWX 710-541-0431

LIST OF ADVERTISERS

RS Number	Advertiser	Page
56	ABM Products	212
452	A M Electronics	124, 125
445	AT-80	115
146	Audio Video Systems	115, 215
229	Access Unlimited	209
34	Acorn Software Products Inc.	254
185	Advance Access Group Inc.	62
260	Advanced Management Strategies Inc.	49
97	Adventure International	134
222	Adventure International	134
224	Adventure International	134
387	Aerocomp, Inc.	216, 217
69	Alpha Byte Storage	31, 45
208	Alpha Products Co.	33
209	Alpha Products Co.	37
210	Alpha Products Co.	35
124	Alphanetics	251
138	Alternate Source, The	264, 292
396	American Business Computers	260
520	American Business Computers	260
521	American Business Computers	260
522	American Business Computers	260
523	American Business Computers	260
524	American Business Computers	261
525	American Business Computers	261
526	American Business Computers	261
527	American Business Computers	261
528	American Business Computers	261
529	American Business Computers	262
227	American Computer Development	188
461	Ancie Labs	233
264	Apparat, Inc.	47, 239
47	Applied Economic Analysis	81
*	Archbold Electronics	237
491	Arcsoft	272
344	A-T Enterprises	64
*	Atlanta Computer Festival	279
48	Automated Simulations	131
338	Automated Simulations	60
380	B. T. Enterprises	78
*	BAP$ Software	248
201	Barstrann Corp.	94
49	Basics & Beyond, Inc.	162
186	Baysian Investment Service	118
237	BCCompco	289
184	Belden Corp.	62
351	Berg Works	222
357	Big Five Software Co.	139
76	Breeze Computing Inc.	57
382	Business Micro Products	76
145	C & S Electronics	198
38	Case Computer Products Co.	72
62	Cecdat, Inc.	121, 207
117	Chicatrug News	266
32	Cload Magazine	257
502	Cload Magazine	257
263	Communications Electronics	302
496	Complete Computer Services	63
100	CompuCover	100
206	Computer Accessory Tech.	260
107	Computer Applications Unlimited	253
349	Computer Applications Unlimited	60
501	Computer Book Club, The	65
199	Computer Case Company	138
372	Computer Discounts of Amer.	283
240	Computer Forms	215
22	Computer Information Exchange	117
130	Computer Plus	277
36	Computer Sales & Service	215
212	Computer Shopper	198
111	Computer Store, The	205
9	Computronics, Inc.	148, 149, 150, 151, 152, 153, 154, 155
498	Compuvest, Inc.	215
204	Comsoft	251
497	Connecticut Microcomputers	159
10	Contract Services Assoc.	249
52	Coolidge & Assoc.	73
78	Coosol, Inc.	105
465	Cornsoft Group	136
378	Cornucopia Software	51
99	Cornwall Computer Systems	276
233	Cottage Software	275
298	CPU Shop, The	111
484	Culpeper Computer Designs	289
121	Custom Electronics	116
259	Cyber Innovations	215
*	Cybernetics Inc.	144
33	Data Resources Corp.	19
274	Data Trans	119
407	Datasoft Inc.	223
234	Diaxis Computer Group, Ltd.	284
226	DigiByte	71
368	Digital Systems Eng.	201, 220
440	Discount Software Group	168
*	Disk Supply Co.	259
88	Documan Software	110
242	Dynamic Software	284
5	E-Z Software	135
278	ESI Lynx	75
197	Echo Products	278
477	EDU-Ware	135
510	Eigen Systems	132
	80 MICROCOMPUTING	146, 179, 190, 191, 171, 231, 235, 293, 299, 300, 301
58	Electronic Specialists	297
328	Electronic Specialists	60
487	Epsilon Systems	248
326	Epson America	61
404	Epson America	103
339	Ever Roll Specialties Co.	61
3	Exatron	C IV
141	FEC Ltd.	132
12	FMG Corp.	163
225	Fantastic Software	266
39	G & L Software Enterprises	200
203	G P Assoc.	235
79	Allen Gelder Software	74, 241
331	Genisco Technology Corp.	64
75	Godbout Electronics	295
281	Golden Braid Software	214
218	Good-Lyddon Data Systems	110
284	Gosub Software	63
481	J. D. Gramm	214
335	Group Technology Ltd.	64
223	H. & H. Trading Company	220
419	Hayden Book Company, Inc.	252
125	Hexagon Systems	41, 193
23	Hobby World Electronics	304
401	Holmes Engineering	272
103	Howe Software	128
37	IJG Inc.	242, 243
2	ISI	143, 182, 183, 184, 185, 230, 237
408	ISI	210
409	ISI	272
410	ISI	289
411	ISI	53
412	ISI	53
413	ISI	53
300	Information Technology Systems	210
327	Inotec, Inc.	62
305	Insiders Software Consultants Inc.	177
246	Interface, Inc.	264
319	Interface Technology	278
287	Interlude	187
485	Interpro	207
190	J P C Products	195
193	Joe Computer	275
85	Johnson Associates	297
432	Keyline Computer Products	221
490	King Electronics	286
149	Kogyosha Company Ltd.	222
375	Krell Software	165
53	LNW Research	189, 201
11	LT Data	227
198	Learning Place, The	82
14	Level IV Products, Inc.	271
391	Lindbergh Systems	159
15	Lobo Drives International	C III
*	MTI	156, 213
451	MTS Enterprises	279
268	Maine Software	295
509	Management Services	116
87	Management Systems Software, Inc.	164
503	McClintock Corp.	119
128	Med Systems Software	172, 173
421	Medfield Computer Software	251
272	Mega Systems	210
104	Mercer Systems, Inc.	279
30	Meta Logix Corp.	13
26	Meta Resources Corp.	9
20	Meta Technologies Corp.	15, 17, 23, 27
27	Metatronics Corp.	11
54	Micro Architect	181
333	Micro Architect Inc.	61
205	Micro Blajak Systems Inc.	100
379	Micro-Design	275
350	Micro-80, Inc.	64
476	Micro-80	251
89	Micro Learningware	100
72	Micro Management Systems Inc.	281
68	Micro Matrix	245
29	Micro Mega	258
310	Micro Mint	115, 292
493	Micro Systems Software	292
182	Micro Systems Software Inc.	64
384	Micro Systems Software Inc.	6
306	Microperipheral Corp.	297
360	Microtek, Inc.	55
109	Micro Works, The	275
8	Midwest Computer Peripheral	303
112	Miller Microcomputer Services	80
24	Mini Micro Mart Inc.	305
98	Minis 'N Micros Inc.	116
221	MISOSYS	126
144	Mumford Micro Systems	267
135	NDM Designs	138
194	New England Business Service, Inc.	193
343	NIK International Trading Inc.	61
245	Okidata Corp.	83
337	Okidata Corp.	61
389	Omega Sales	42
357	Omikron	57
296	Orange Micro Inc.	97
376	Orion Instruments	126
127	P C Newsletter	284
370	Pacific Exchanges	132
153	Pacific Office Systems	164
64	Pan American Electronics	176
207	Pensadyne Computer Services	74, 230
1	Percom Data Company, Inc.	C II
258	Percom Data Company, Inc.	3
336	Percom Data Company	60
512	Peripherals Plus	264
51	Perry Gas & Oil	142
422	Personal Micro Computers, Inc.	123
*	Phase I Systems	196
334	Phase One Systems	61
131	Plus Computer Technology, Inc.	175
507	Pro 80 Systems	169
17	Program Store, The/Realsoft	66, 67
364	Programmer's Guild, The	264
277	Progressive Electronics	280
441	Prosoft	73
269	Quant Systems	105
304	Quarp Publishing	82
41	Racet Computes	285
492	Ram Intelligence	286
236	Rand's Inc.	101
*	Realty Software Company	94
70	Remsoft Inc.	133
499	Remtron	248
276	Richcraft Engineering Ltd.	94
468	Rochester Data Inc.	200
373	SID	207
143	Sales Data, Inc.	235
329	Howard W. Sams and Co., Inc.	61, 64
341	Alan Saville	62
291	Scientific Engineering Lab.	110
504	See-Thru Enterprises	63
50	Select Info. Sys. Inc.	229
345	Serpent Software	60
297	Service Technologies Inc.	298
19	Simutek	199
91	68 Micro Journal	215
183	Small Business Systems Group Inc.	62
*	Snapp Inc.	28, 29
434	Soft Sector Marketing Inc.	107
495	Soft Sector Marketing Inc.	106
356	Software Affair	63
299	Software Concepts	263
325	Software Critic	62
330	Software Etc.	61
13	Software House International	119
289	Softworx, Inc.	238
132	Spectral Associates	234
275	Speedway Electronics	266
511	Starbuck Data Co.	73
438	Stocking Source, The	145
93	Strawberry Software, Inc.	170
*	Strubie, William	128
82	Sturdivant & Dunn, Inc.	121
150	Sublogic	129
267	Suma Microware	215
151	Sun Research	171
403	Superior Software	132
*	Synergetic Solutions	138
231	Synergistic Solar, Inc.	162
*	Syntex Electronic Innovations	118
358	Syracuse R & D Center	120
148	Tab Sales Company	279
280	Tape Tronics	276
45	Taranto & Associates	59
143	Task Computer Applications	113
147	Task Computer Applications	113
342	Teach Yourself By Computer Software	62
25	Texas Computer Systems	231
392	Texas Digital Inc.	218
494	Used Computer Annex	235
*	V R Data Corp.	291
137	Van Horn Office Supply	125
65	Vista Computer Co.	93
332	Westico, Inc.	82
346	Westico, Inc.	64
347	Westico, Inc.	60
348	Westico, Inc.	54
279	Wilson Software Division	110
482	York 10 Computerware	276

*This advertiser prefers to be contacted directly.

READER SERVICE

This card valid until August 31, 1981

Please help us to bring you a better magazine—by answering these questions.

I. What is your age?
- ☐ A. under 18
- ☐ B. 18-22
- ☐ C. 23-40
- ☐ D. 41-60
- ☐ E. over 60

II. What is your occupation?
- ☐ 1. Professional
- ☐ 2. Engineer
- ☐ 3. Data processing
- ☐ 4. Business
- ☐ 5. Education
- ☐ 6. Technician
- ☐ 7. Student
- ☐ 8. Other

III. What are your primary applications of your TRS-80 (check only two)?
- ☐ A. Business
- ☐ B. Games
- ☐ C. Home
- ☐ D. Education
- ☐ E. Scientific
- ☐ F. Control
- ☐ G. Music

IV. Your TRS-80, is it a
- ☐ 1. Level I
- ☐ 2. Level II
- ☐ 3. Model II
- ☐ 4. Don't own one yet

V. What peripherals do you have (check all that apply)?
- ☐ A. Expansion interface
- ☐ B. Disk
- ☐ C. Printer

VI. How much have you spent on hardware?
- ☐ 1. less than $500
- ☐ 2. $500-1,000
- ☐ 3. $1,000-2,000
- ☐ 4. $2,000-4,000
- ☐ 5. $4,000-6,000
- ☐ 6. more than $6,000

VII. How much have you spent on software?
- ☐ A. less than $100
- ☐ B. $100-250
- ☐ C. $250-500
- ☐ D. $500-1,000
- ☐ E. more than $1,000

VIII. What is your level of education?
- ☐ 1. Post-graduate
- ☐ 2. College
- ☐ 3. High school

IX. How many people read your copy of 80 Microcomputing?
- ☐ A. 1
- ☐ B. 2
- ☐ C. 3
- ☐ D. 4 or more

X. If you are not a subscriber please circle number 500.

80 Microcomputing • POB 2743 • Clinton IA 52735

Reader Service: To receive more information from any of the advertisers in this issue of *80 Microcomputing*, circle the number on the postage-paid Reader Service Card that corresponds with the Reader Service number on the ad in which you are interested. You will find numbers, preceded by a ✓, near the logo of each advertiser. Complete the entire card, drop into a mailbox and in 4-6 weeks you will hear from the advertiser directly.

1	6	11	16	21	151	156	161	166	171	301	306	311	316	321	451	456	461	466	471
2	7	12	17	22	152	157	162	167	172	302	307	312	317	322	452	457	462	467	472
3	8	13	18	23	153	158	163	168	173	303	308	313	318	323	453	458	463	468	473
4	9	14	19	24	154	159	164	169	174	304	309	314	319	324	454	459	464	469	474
5	10	15	20	25	155	160	165	170	175	305	310	315	320	325	455	460	465	470	475
26	31	36	41	46	176	181	186	191	196	326	331	336	341	346	476	481	486	491	496
27	32	37	42	47	177	182	187	192	197	327	332	337	342	347	477	482	487	492	497
28	33	38	43	48	178	183	188	193	198	328	333	338	343	348	478	483	488	493	498
29	34	39	44	49	179	184	189	194	199	329	334	339	344	349	479	484	489	494	499
30	35	40	45	50	180	185	190	195	200	330	335	340	345	350	480	485	490	495	500
51	56	61	66	71	201	206	211	216	221	351	356	361	366	371	501	506	511	516	521
52	57	62	67	72	202	207	212	217	222	352	357	362	367	372	502	507	512	517	522
53	58	63	68	73	203	208	213	218	223	353	358	363	368	373	503	508	513	518	523
54	59	64	69	74	204	209	214	219	224	354	359	364	369	374	504	509	514	519	524
55	60	65	70	75	205	210	215	220	225	355	360	365	370	375	505	510	515	520	525
76	81	86	91	96	226	231	236	241	246	376	381	386	391	396	526	531	536	541	546
77	82	87	92	97	227	232	237	242	247	377	382	387	392	397	527	532	537	542	547
78	83	88	93	98	228	233	238	243	248	378	383	388	393	398	528	533	538	543	548
79	84	89	94	99	229	234	239	244	249	379	384	389	394	399	529	534	539	544	549
80	85	90	95	100	230	235	240	245	250	380	385	390	395	400	530	535	540	545	550
101	106	111	116	121	251	256	261	266	271	401	406	411	416	421	551	556	561	566	571
102	107	112	117	122	252	257	262	267	272	402	407	412	417	422	552	557	562	567	572
103	108	113	118	123	253	258	263	268	273	403	408	413	418	423	553	558	563	568	573
104	109	114	119	124	254	259	264	269	274	404	409	414	419	424	554	559	564	569	574
105	110	115	120	125	255	260	265	270	275	405	410	415	420	425	555	560	565	570	575
126	131	136	141	146	276	281	286	291	296	426	431	436	441	446	576	581	586	591	596
127	132	137	142	147	277	282	287	292	297	427	432	437	442	447	577	582	587	592	597
128	133	138	143	148	278	283	288	293	298	428	433	438	443	448	578	583	588	593	598
129	134	139	144	149	279	284	289	294	299	429	434	439	444	449	579	584	589	594	599
130	135	140	145	150	280	285	290	295	300	430	435	440	445	450	580	585	590	595	600

Name ____________________

Address ____________________

City ____________________ State ______ Zip ______

BOOKS

80 Microcomputing
Peterborough NH 03458

Please send me the following 80 Microcomputing products:

Qty.	Catalog #	Title	Unit Price	Total

Add $1 shipping/handling ______

Total ______

Enclosed $______ ☐ Check ☐ M.O.
Bill: ☐ AE ☐ MC ☐ VISA

Card # ____________________

Exp date ______ Interbank # ______

Signature ____________________

Name ____________________

Address ____________________

City ______ State ______ Zip ______

Please allow 4–6 weeks for delivery

No C.O.D. orders accepted.

6-81

SUBSCRIPTION

80 Microcomputing POB 981
Farmingdale NY 11737

80 Microcomputing subscribers SAVE 40% Subscribe now.

Please allow 4–6 weeks for delivery

☐ New Subscription ☐ Renewal
☐ 1 year—$18 ☐ 2 years—$30 ☐ 3 years—$45
☐ Enclosed $______ ☐ Check ☐ M.O.
Bill: ☐ MC ☐ VISA ☐ AE ☐ Me

Card # ______ Exp. date ______

Signature ______ Interbank # ______

Name ____________________

Address ____________________

City ______ State ______ Zip ______

Canada: $20 1 year only, US funds
Other foreign: $28 one year only, US funds

316B9

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 217 CLINTON IA 52735

POSTAGE WILL BE PAID BY ADDRESSEE

80 microcomputing T.M.

POB 2743
Clinton IA 52735

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 780 FARMINGDALE NY 11737

POSTAGE WILL BE PAID BY ADDRESSEE

80 microcomputing T.M.

Subscription Dept.
POB 981
Farmingdale NY 11737

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 80 PETERBOROUGH NH 03458

POSTAGE WILL BE PAID BY ADDRESSEE

80 microcomputing T.M.

Peterborough NH 03458

Att. Mail Order

Expand your TRS-80*

With The LOBO LX-80 Expansion Interface

Now you can realize all the power and potential of your TRS-80*, Model 1. If it's add-on memory you need, your LX-80 can accommodate up to four 5¼-inch, single- or double-density 35, 40 or 80 track mini-floppies, four 8-inch floppies (single or double sided), and up to eight Winchester fixed disk-drives (5¼". 8", 14").

LOBO's powerful new LDOS™ operating system, provided with your LX-80, allows for the use of any eight drives, in any combination, single or double density.

And there's more ... lots more. There are two parallel ports (standard), two serial ports (optional), a keyboard ROM override switch, and a 32K memory expansion (optional). Send for a free LX-80/TRS-80 cost performance comparison chart.

For the full story on how the LX-80 can expand your TRS-80, see your nearest LOBO dealer, or write or call:

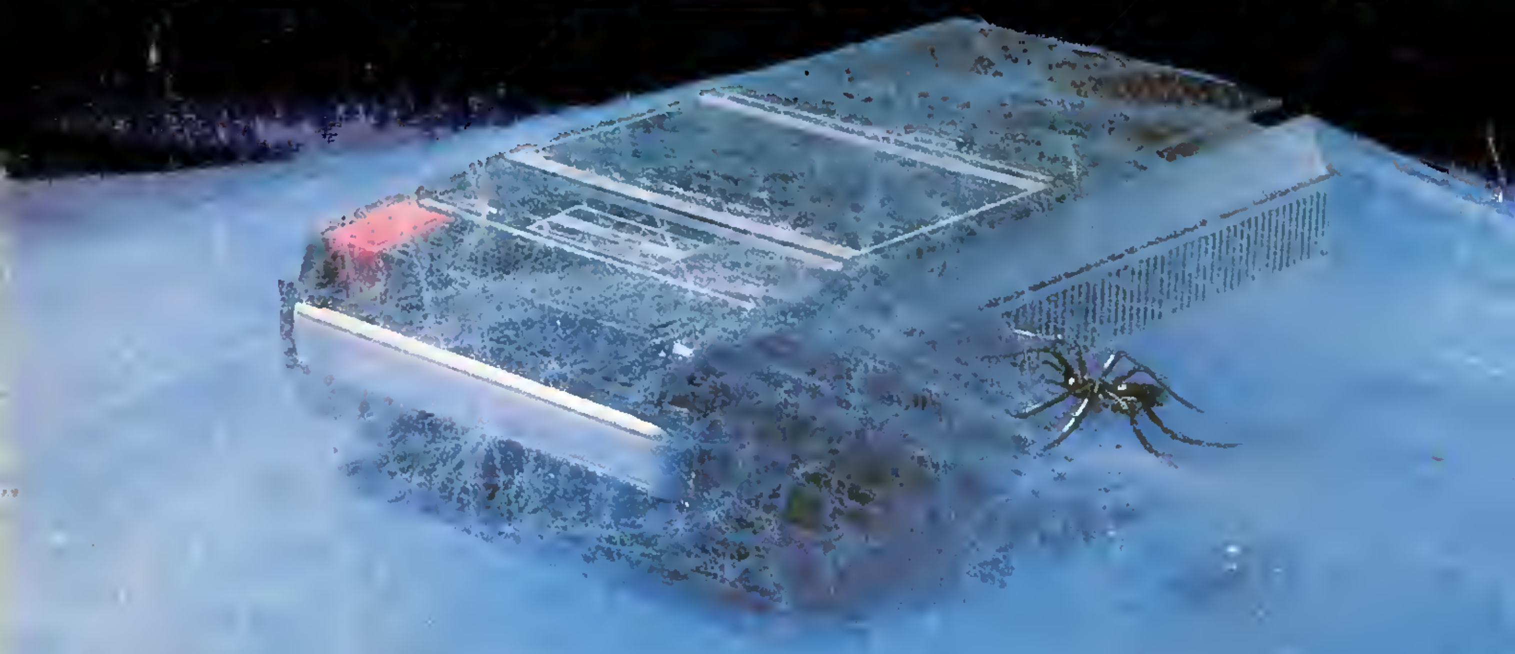
*TRS-80 is a registered trademark of Radio Shack, a Tandy Company.

LOBO drives INTERNATIONAL

LOBO DRIVES INT'L
354 South Fairview Ave.
Goleta, CA 93117
(805) 683-1576

15

TIRED OF WAITING?



Frustrating isn't it! No matter how much you speed up your program it still seems to take forever to save data onto a cassette. Wouldn't it be great if someone could design a mass storage system with the speed of a disk, but at half the cost? *Exatron* did, the *Exatron Stringy Floppy (ESF)* .

Totally self-contained, the ESF is an extremely fast, reliable, and economical alternative to cassette or disk storage of programs or data. All of the ESF's operations are under the computer's control, with no buttons, switches, knobs or levers to adjust or forget.

The ESF uses a miniature tape cartridge, about the size of a business card, called a wafer. The transport mechanism uses a direct drive motor with only one moving part. Designed to read and write digital data only, the ESF suffers from none of the drawbacks of cassettes - without the expense of disks.

Several versions of the ESF are available, for the *TRS-80, Apple, PET, OSI* and an *RS 232* unit. Even the slowest of the units is 15 times faster than a cassette, and all are as reliable as disk drives - in fact a lot of users say they are *more* reliable!

excellence in electronics

exatron

To get further information about the ESF give Exatron a call on their Hot Line *800-538 8559* (inside California 408-737 7111).

If you can't wait any longer then take advantage of their 30 day money-back guarantee, you've nothing to lose but time!

181 Commercial Street
Sunnyvale, CA 94086

3

TRS-80, Apple and PET are trademarks of Tandy, Apple and Commodore respectively.